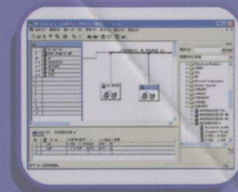
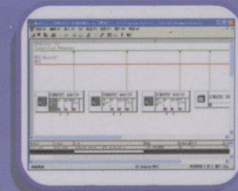
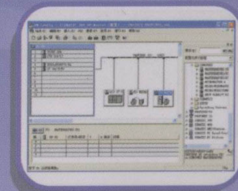

自动化技术轻松入门丛书

S7-300/400 PLC 基础与案例精选

主　编　向晓汉
副主编　陆　彬
参　编　李润海　刘遥遥





本书附有配套资源
下载网址：www.cmpbook.com

- 免费下载重点实例源程序
- 免费下载操作过程视频文件


机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE Caliente.COM

自动化技术轻松入门丛书

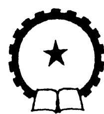
S7-300/400 PLC 基础与案例精选

主　编　向晓汉

副主编　陆　彬

参　编　李润海　刘遥遥

主　审　郭　琼



机 械 工 业 出 版 社

本书从基础和实用出发，涵盖的主要内容包括 S7-300/400 PLC、变频器和组态软件。全书分两个部分：第一部分为入门学习，主要介绍 S7-300/400 PLC 的硬件和接线、STEP 7 软件的使用、PLC 的编程语言、编程方法与调试；第二部分为实例应用，包括 PLC 的通信、PLC 在过程控制的应用、PLC 在变频调速中的应用和上位机对 PLC 的监控。

本书内容丰富，重点突出，强调知识的实用性，几乎每章中都配有大量实用的例题，便于读者模仿学习，另外每章配有习题供读者练习。大部分实例都有详细的软件、硬件配置清单，并配有接线图和程序。本书的配套资源中有重点内容的程序和操作视频资料。

本书可以作为工程技术人员学习 S7-300/400 PLC 入门和提高级用书，也可以作为大中专院校机电类、信息类专业的教材。

图书在版编目（CIP）数据

S7-300/400 PLC 基础与案例精选 / 向晓汉主编. —北京：机械工业出版社，2010.10 （2011.7重印）
（自动化技术轻松入门丛书）
ISBN 978-7-111-32002-9

Ⅰ. ①S… Ⅱ. ①向… Ⅲ. ①可编程序控制器 Ⅳ. ①TM571.6

中国版本图书馆 CIP 数据核字（2010）第 186635 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：时 静
责任编辑：时 静 黄 伟
责任印制：李 妍

北京振兴源印务有限公司印刷

2011 年 7 月第 1 版 • 第 2 次印刷
184mm×260mm • 21.25 印张 • 524 千字
3501－6500册
标准书号：ISBN 978-7-111-32002-9
定价：40.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
电话服务
社服务中心：（010）88361066
销 售 一 部：（010）68326294
销 售 二 部：（010）88379649
读者服务部：（010）68993821

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

封面无防伪标均为盗版

前　　言

随着计算机技术的发展，以可编程序控制器（PLC）、变频器、计算机通信和组态软件等为主体的新型电气控制系统已经逐渐取代传统的继电器电气控制系统，并广泛应用于各行业。其中西门子 PLC 具有卓越的性能，在工控市场占有非常大的份额，应用十分广泛。虽然 S7-300/400 PLC 被大多数技术人员接受，但长期以来，S7-300/400 PLC 一直公认是比较难入门的。为了使读者能更好地掌握相关知识，我们在总结长期的教学经验和工程实践的基础上，联合相关企业人员，共同编写了本书，力争使读者通过“看书”就能学会 S7-300/400 PLC。本书将力求简单和详细，用较多的小例子引领读者入门，让读者读完入门部分后，能完成简单的工程。应用部分精选工程的实际案例，供读者模仿学习，提高读者解决实际问题的能力。

我们在编写过程中，将一些生动的操作实例融入该书，以提高读者的学习兴趣。本书与其他相关书籍相比，具有以下特点。

（1）用实例引导读者学习。该书的大部分章节用精选的例子讲解。例如，用例子说明现场总线通信实现的全过程。

（2）重点的例子都包含软硬件的配置方案图、接线图和程序，而且为确保程序的正确性，本书中程序已经在 PLC 上运行通过。

（3）对于比较复杂的例子，为了便于读者学习，均配有视频。如工业以太网通信的硬件组态，就配有编者的组态过程视频。

（4）该书实例易于工程移植。

全书共分 9 章。第 1、4、5、6 章由无锡职业技术学院的向晓汉编写；第 2、3、8 章由无锡雷华科技有限公司的陆彬编写；第 7 章由无锡雪浪输送机厂的刘遥遥编写；第 9 章由无锡小天鹅股份有限公司的李润海编写。本书由向晓汉任主编，陆彬任副主编。郭琼副教授任主审。

由于编者水平有限，缺点和错误在所难免，敬请读者批评指正，编者将万分感激！

作 者

目　录

前言
第 1 章　可编程序控制器（PLC）基础……1
1.1　概述……1
1.1.1　PLC 的发展历史……1
1.1.2　PLC 的主要特点……2
1.1.3　PLC 的应用范围……3
1.1.4　PLC 的分类与性能指标……3
1.1.5　PLC 与继电器系统的比较……4
1.1.6　PLC 与微机的比较……5
1.1.7　PLC 的发展趋势……5
1.1.8　PLC 在我国的使用情况……5
1.2　可编程序控制器的结构和工作原理……6
1.2.1　可编程序控制器的硬件组成……6
1.2.2　可编程序控制器的工作原理……9
1.2.3　可编程序控制器的立即输入、输出功能……10
小结……11
习题……11
第 2 章　S7-300/400 PLC 的硬件介绍……12
2.1　S7-300 PLC 常用模块及其接线……12
2.1.1　西门子 PLC 简介……12
2.1.2　S7-300 PLC 常用模块及其接线……13
2.2　S7- 400 PLC 常用模块及其接线……29
2.2.1　S7-400 PLC 的概述……29
2.2.2　S7-400 PLC 的机架……30
2.2.3　S7-400 PLC 常用模块及其连接……32
小结……42
习题……42
第 3 章　STEP 7 软件使用入门……44
3.1　STEP 7 简介……44
3.1.1　初识 STEP 7……44
3.1.2　安装 STEP 7 注意事项……45
3.2　编程界面的 SIMATIC 管理器……45
3.2.1　创建项目……45





3.2.2　编辑项目······47
3.3　硬件组态与参数设置······49
3.3.1　硬件组态······49
3.3.2　参数设置······52
3.3.3　硬件的更新和 GSD 文件安装······60
3.4　下载和上传······62
3.4.1　下载······62
3.4.2　上传······66
3.5　软件编程······67
3.6　打印和归档······69
3.6.1　打印······69
3.6.2　归档······70
3.7　用STEP 7 V5.4 建立一个完整的项目······71
3.8　使用帮助······76
3.8.1　查找关键字或者功能······76
3.8.2　了解某个逻辑块 FB/SFB/FC/SFC 的功能及管脚的定义······76
小结······77
习题······78
第 4 章　S7-300/400 PLC 的编程语言······79
4.1　S7-300/400 PLC 的编程基础知识······79
4.1.1　编程元件与数据类型······79
4.1.2　寻址方式······82
4.1.3　编程语言······86
4.2　CPU 中的寄存器······87
4.2.1　累加器（ACCUx）······87
4.2.2　状态字寄存器（16 位）······88
4.2.3　数据块寄存器······88
4.3　位逻辑指令······88
4.4　定时器与计数器指令······95
4.4.1　定时器······95
4.4.2　计数器······104
4.5　数据处理与运算指令······107
4.5.1　装载与传送指令······107
4.5.2　比较指令······111
4.5.3　转换指令······114
4.5.4　移位与循环指令······120
4.5.5　算术运算指令······125
4.5.6　控制指令······134
4.6　S7-300/400 PLC 的程序结构······137



4.6.1 功能、功能块和数据块……137
4.6.2 组织块（OB）……147
4.6.3 STL 源文件……162
4.7 实例……167
小结……178
习题……179
第 5 章 S7-300/400 PLC 的编程方法与调试……181
5.1 功能图……181
5.1.1 功能图的画法……181
5.1.2 梯形图编程的原则……185
5.2 逻辑控制的梯形图编程方法……187
5.2.1 经验设计法……187
5.2.2 功能图设计法……189
5.3 S7-300/400 PCL 的诊断与调试方法……195
5.3.1 使用状态和出错 LED 进行诊断……196
5.3.2 使用 STEP 7 的软件诊断功能进行硬件诊断……196
5.3.3 用变量监控表进行调试……199
5.3.4 使用 PLCSIM 软件进行调试……202
5.4 实例……205
小结……212
习题……213
第 6 章 S7-300/400 PLC 的通信及其应用……215
6.1 通信基础知识……215
6.1.1 通信的基本概念……215
6.1.2 RS-485 标准串行接口……217
6.1.3 OSI 参考模型……218
6.2 MPI 通信及其应用……219
6.2.1 MPI 通信简介……219
6.2.2 S7-200 PLC 与 S7-300 PLC 间的 MPI 通信……220
6.2.3 S7-300 PLC 与 S7-300 PLC 间的 MPI 通信……227
6.2.4 S7-300/400 PLC 与 S7-400 PLC 间的 MPI 通信……236
6.3 PROFIBUS 通信及其应用……240
6.3.1 PROFIBUS 通信基础……240
6.3.2 S7-200 PLC 与 S7-300 PLC 间的现场总线通信……242
6.3.3 S7-300 PLC 与 S7-300 PLC 间的现场总线通信……253
6.4 以太网通信及其应用……262
6.4.1 以太网通信基础……262
6.4.2 S7-200 PLC 与 S7-300 PLC 间的以太网通信……265
6.4.3 S7-300 PLC 间的以太网通信……272





小结 ……280

习题 ……281

第 7 章　S7-300 PLC 在电炉温度控制系统中的应用 ……282

7.1　PID 控制简介 ……282

7.2　利用 S7-300 PLC 进行电炉的温度控制……284

小结 ……289

习题 ……289

第 8 章　S7-300 PLC 在模块化生产线变频调速系统中的应用 ……291

8.1　认识模块化生产线的运输站 ……291

8.2　西门子 MM440 变频器使用简介 ……291

8.2.1　认识变频器 ……291

8.2.2　西门子 MM440 变频器使用简介……293

8.3　运输站的控制……297

8.3.1　运输站变频器的 BOP 调速……297

8.3.2　运输站变频器的模拟量调速……298

8.3.3　运输站变频器的多段调速 ……300

8.3.4　运输站变频器的通信调速 ……303

小结 ……313

习题 ……313

第 9 章　上位机对 S7-300/400 PLC 的监控……314

9.1　简单组态软件工程的建立 ……314

9.1.1　认识组态软件……314

9.1.2　建立工程……316

9.2　搬运站组态工程的建立……323

9.2.1　变量 ……323

9.2.2　动画相关……324

9.2.3　命令语言程序……326

9.2.4　创建搬运站工程……329

小结 ……330

习题 ……330

参考文献……331

第 1 章

可编程序控制器（PLC）基础

本章介绍可编程序控制器的历史、功能、特点、应用范围、发展趋势、在我国的使用情况、其结构和工作原理等知识，使读者初步了解可编程序控制器，为学习本书后续内容做必要准备。

1.1 概述

可编程序控制器（Programmable Logic Controller）简称 PLC，国际电工委员会（IEC）于 1985 年对可编程序控制器作了如下定义："可编程序控制器是一种数字运算操作的电子系统，专为在工业环境下应用而设计。它采用可编程序的存储器，用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字、模拟的输入和输出，控制各种类型的机械或生产过程。可编程序控制器及其有关设备，都应按易于与工业控制系统连成一个整体，易于扩充功能的原则设计。PLC 是一种工业计算机，其种类繁多，不同厂家的产品有其各自的特点，但作为工业标准设备，可编程序控制器又有一定的共性。"

1.1.1 PLC的发展历史

20 世纪 60 年代以前，汽车生产线的自动控制系统基本上都由继电器控制装置构成。当时每次改型都直接导致继电器控制装置的重新设计和安装，福特汽车公司的老板曾经说，无论顾客需要什么样的汽车，福特汽车永远是黑色的，从侧面反映汽车改型和升级换代比较困难。为了改变这一状况，1969 年，美国通用汽车公司（GM）公开招标，要求用新的装置取代继电器控制装置，并提出十项招标指标，要求编程方便、现场可修改程序、维修方便、采用模块化设计、体积小、可与计算机通信等。同年，美国数字设备公司（DEC）研制出了世界上第一台可编程序控制器 PDP-14，在美国通用汽车公司的生产线上试用成功，并取得了满意的效果，可编程序控制器从此诞生。由于当时的 PLC 只能取代继电器接触器控制，功能仅限于逻辑运算、计时、计数等，所以称为"可编程逻辑控制器"。伴随着微电子技术、控制技术与信息技术的不断发展，可编程序控制器的功能不断增强。美国电气制造商协会（NEMA）于 1980 年正式将其命名为"可编程控制器"，简称 PC。由于这个名称和个人计算机的简称相同，容易混淆，因此在我国，很多人仍然习惯称可编程序控制器为 PLC。

由于 PLC 具有易学易用、操作方便、可靠性高、体积小、通用灵活以及使用寿命长等一系列优点。因此，很快 PLC 就在工业中得到了广泛的应用。同时，这一新技术也受到其他国家的重视。1971 年日本引进这项技术，很快研制出日本第一台 PLC，欧洲于 1973 年研制出第一台 PLC，我国从 1974 年开始研制，1977 年国产 PLC 正式投入工业应用。



进入20世纪80年代以来，随着电子技术的迅猛发展，以16位和32位微处理器构成的微机化 PLC 得到快速发展，（例如 GE-FANUC 的 RX7i，使用的是赛扬 CPU，其主频达1GHz，信息处理能力几乎和个人电脑相当）使得 PLC 在设计、性能价格比以及应用方面有了突破，不仅控制功能增强，功耗和体积减小，成本下降，可靠性提高，编程和故障检测更为灵活方便，而且随着远程I/O和通信网络、数据处理和图像显示的发展，已经使得PLC普遍用于控制复杂生产过程。PLC目前已经成为工厂自动化的三大支柱之一。

1.1.2 PLC的主要特点

PLC之所以高速发展，除了工业自动化的客观需要外，还有许多适合工业控制的独特优点，它较好地解决了工业控制领域中普遍关心的可靠、安全、灵活、方便、经济等问题。其主要特点如下：

1．抗干扰能力强，可靠性高

在传统的继电器控制系统中，使用了大量的中间继电器、时间继电器，由于器件的固有缺点，如器件老化、接触不良、触点抖动等现象，大大降低了系统的可靠性。而在 PLC 控制系统中大量的开关动作由无触点的半导体电路完成，因此故障大大减少。

此外，PLC的硬件和软件方面采取了措施，提高其可靠性。在硬件方面，所有的I/O接口都采用了光电隔离，使得外部电路与 PLC 内部电路实现了物理隔离。各模块都采用了屏蔽措施，以防止辐射干扰。电路中采用了滤波技术，以防止或抑制高频干扰。在软件方面，PLC 具有良好的自诊断功能，一旦系统的软硬件发生异常情况，CPU 会立即采取有效措施，防止故障扩大。通常PLC具有看门狗功能。

对于大型的PLC系统，还可以采用双CPU构成冗余系统或者3 CPU构成表决系统，使系统的可靠性进一步提高。

2．程序简单易学，系统的设计调试周期短

PLC是面向用户的设备，PLC的生产厂家充分考虑到现场技术人员的技能和习惯，可采用梯形图或面向工业控制的简单指令形式。梯形图与继电器原理图很相似，直观、易懂、易掌握，不需要学习专门的计算机知识和语言。设计人员可以在设计室设计、修改和模拟调试程序，非常方便。

3．安装简单，维修方便

PLC不需要专门的机房，可以在各种工业环境下直接运行，使用时只需将现场的各种设备与PLC相应的I/O端相连接，即可投入运行。各种模块上均有运行和故障指示装置，便于用户了解运行情况和查找故障。

4．采用模块化结构，体积小，重量轻

为了适应工业控制需求，除了整体式 PLC 外，绝大多数 PLC 采用模块化结构。PLC 的各部件，包括 CPU、电源、I/O 接口等都采用模块化设计。此外，PLC 相对于通用工控机，其体积和重量都要小得多。

5．丰富的I/O接口模块，扩展能力强

PLC针对不同的工业现场信号（如交流或直流、开关量或模拟量、电压或电流、脉冲或电位、强电或弱电等）有相应的 I/O 模块与工业现场的器件或设备（如按钮、行程开关、接近开关、传感器及变送器、电磁线圈、控制阀等）直接连接。另外，为了提高操作

性能，它还有多种人–机对话的接口模块，为了组成工业局部网络，它还有多种通信联网的接口模块等。

1.1.3　PLC的应用范围

目前，PLC 在国内外已广泛应用于机床、控制系统、自动化楼宇、钢铁、石油、化工、电力、建材、汽车、纺织机械、交通运输、环保以及文化娱乐等各行各业。随着 PLC 性能价格比的不断提高，应用范围还将不断扩大，应用场合可以说是无处不在，具体应用大致可归纳为以下几类：

1．顺序控制

这是 PLC 最基本、最广泛应用的领域，它取代传统的继电器顺序控制，PLC 用于单机控制、多机群控制、自动化生产线的控制。例如数控机床、注塑机、印刷机械、电梯控制和纺织机械等。

2．计数和定时控制

PLC 为用户提供了足够的定时器和计数器，并设置相关的定时和计数指令，PLC 的计数器和定时器精度高、使用方便，可以取代继电器系统中的时间继电器和计数器。

3．位置控制

大多数的 PLC 制造商，目前都提供拖动步进电动机或伺服电动机的单轴或多轴位置控制模块，这一功能可广泛用于各种机械，如金属切削机床、装配机械等。

4．模拟量处理

PLC 通过模拟量的输入/输出模块，实现模拟量与数字量的转换，并对模拟量进行控制，有的还具有 PID 控制功能。例如用于锅炉的水位、压力和温度控制。

5．数据处理

现代的 PLC 具有数学运算、数据传递、转换、排序和查表等功能，也能完成数据的采集、分析和处理。

6．通信联网

PLC 的通信包括 PLC 相互之间、PLC 与上位计算机、PLC 和其他智能设备之间的通信。PLC 系统与通用计算机可以直接或通过通信处理单元、通信转接器相连构成网络，以实现信息的交换，并可构成“集中管理、分散控制”的分布式控制系统，满足工厂自动化系统的需要。

1.1.4　PLC的分类与性能指标

1．PLC的分类

（1）从组成结构形式分类

可以将 PLC 分为两类：一类是整体式 PLC（也称单元式），其特点是电源、中央处理单元、I/O 接口都集成在一个机壳内；另一类是标准模板式结构化 PLC（也称组合式），其特点是电源模板、中央处理单元模板、I/O 接口模板等在结构上是相互独立的，可根据具体的应用要求，选择合适的模块，安装在固定的机架或导轨上，构成一个完整的 PLC 应用系统。

（2）按 I/O 点容量分类

1）小型 PLC。小型 PLC 的 I/O 点数一般在 128 点以下。



2）中型 PLC。中型 PLC 采用模块化结构，其 I/O 点数一般在 256～1024 点之间。

3）大型 PLC。一般 I/O 点数在 1024 点以上的称为大型 PLC。

2．PLC的性能指标

各厂家的 PLC 虽然各有特色，但其主要性能指标是相同的。

（1）输入/输出（I/O）点数

输入/输出（I/O）点数是最重要的一项技术指标，也是 PLC 选型时最重要的指标 PLC 的面板上连接外部输入、输出端子数常称为“点数”，用输入与输出点数的和表示。点数越多表示 PLC 可接入的输入器件和输出器件越多，控制规模越大。

（2）扫描速度

扫描速度是指 PLC 执行程序的速度，以 ms/K 步为单位，即执行 1K 步指令所需的时间。l 步占 1 个地址单元。

（3）存储容量

存储容量通常用千字（KW）或千字节（KB）、千位（Kbit）来表示。这里 1K＝1024。有的 PLC 用“步”来衡量，一步占用一个地址单元。存储容量表示 PLC 能存放多少用户程序。例如，三菱型号为 FX2N−48MR 的 PLC 存储容量为 8000 步。有的 PLC 的存储容量可以根据需要配置，有的 PLC 的存储器可以扩展。

（4）指令系统

指令系统表示该 PLC 软件功能的强弱。指令越多，编程功能就越强。

（5）内部寄存器（继电器）

PLC 内部有许多寄存器用来存放变量、中间结果、数据等，还有许多辅助寄存器可供用户使用。因此寄存器的配置也是衡量 PLC 功能的一项指标。

（6）扩展能力

扩展能力是反映 PLC 性能的重要指标之一。PLC 除了主控模块外，还可配置实现各种特殊功能的高功能模块。例如 A/D 模块、D/A 模块、高速计数模块、远程通信模块等。

1.1.5　PLC与继电器系统的比较

在 PLC 出现以前，继电器硬接线电路是逻辑、顺序控制的唯一执行者。它结构简单、价格低廉，一直被广泛应用。PLC 出现后，几乎所有的方面都超越继电器控制系统，两者的性能比较见表 1−1。

表 1−1　可编程序控制器与继电器控制系统的比较

序号	比较项目	继电器控制	可编程序控制器控制
1	控制逻辑	硬接线多、体积大、连线多	软逻辑、体积小、接线少、控制灵活
2	控制速度	通过触点开关实现控制，动作受继电器硬件限制，通常超过 10ms	由半导体电路实现控制，指令执行时间段，一般为微秒级
3	定时控制	由时间继电器控制，精度差	由集成电路的定时器完成，精度高
4	设计与施工	设计、施工、调试必须按照顺序进行，周期长	系统设计完成后，施工与程序设计同时进行，周期短
5	可靠性与维护	继电器的触点寿命短，可靠性和维护性差	无触点，寿命长，可靠性高，有自诊断功能
6	价格	价格低	价格高

1.1.6　PLC与微机的比较

采用微电子技术制造的可编程序控制器与微机一样，也由 CPU、ROM（或者 FLASH）、RAM、I/O 接口等组成，但又不同于一般的微机，可编程序控制器采用了特殊的抗干扰技术，是一种特殊的工业控制计算机，更加适合工业控制。两者的性能比较见表 1−2。

表 1−2　PLC 与微机的比较

序号	比较项目	可编程序控制器控制	微机控制
1	应用范围	工业控制	科学计算、数据处理、计算机通信
2	使用环境	工业现场	具有一定温度和湿度的机房
3	输入/输出	控制强电设备，需要隔离	与主机弱电联系，不隔离
4	程序设计	一般使用梯形图语言，易学易用	编程语言丰富，如 C、BASIC 等
5	系统功能	自诊断、监控	使用操作系统
6	工作方式	循环扫描方式和中断方式	中断方式

1.1.7　PLC的发展趋势

1．向高性能、高速度、大容量发展

2．网络化

强化通信能力和网络化，向下将多个可编程序控制器或者多个 I/O 框架相连；向上与工业计算机、以太网等相连，构成整个工厂的自动化控制系统。即便是微型的 S7−200 PLC 也能组成多种网络，通信功能十分强大。

3．小型化、低成本、简单易用

目前，市场上小型 PLC 的价格有的只有几百元人民币。

4．不断提高编程软件的功能

编程软件可以对 PLC 控制系统的硬件组态，在屏幕上可以直接生成和编辑梯形图、指令表、功能块图和顺序功能图程序，并可以实现不同编程语言的相互转换。程序可以下载、存盘和打印，通过网络或电话线，还可以实现远程编程。

5．适合PLC应用的新模块

随着科技的发展，对工业控制领域将提出更高的、更特殊的要求。因此，必须开发特殊功能模块来满足这些要求。

6．PLC的软件化与PC化

目前已有多家厂商推出了在 PC 上运行的可实现 PLC 功能的软件包，也称为“软 PLC”，“软 PLC”的性能价格比比传统的“硬 PLC”更高，是 PLC 的一个发展方向。

PC 化的 PLC 类似于 PLC，但它采用了 PC 的 CPU，功能十分强大，如 GE 的 Rx7i 和 Rx3i 使用的就是工控机用的赛扬 CPU，主频已经达到 1GHz。

1.1.8　PLC在我国的使用情况

目前 PLC 在我国得到了广泛的应用，几乎所有知名厂家的 PLC 在我国都有应用。在我国使用较多的 PLC 品牌有德国的西门子（SIEMENS），法国的施耐德（Schneider），奥地利的贝加莱（B&R），日本的三菱（MITSUBISHI）、欧姆龙（OMRON）、日立（Hitachi）和松



下（Panasonic），美国的通用电气（GE）和罗克维尔（Rockwell）等。而大中型可编程控制器主要以欧美的品牌为主。目前 95%以上的 PLC 市场被国外品牌所占领。

我国自主品牌的 PLC 生产厂家有近 30 余家。在目前已经上市的众多 PLC 产品中，还没有形成规模化的生产和名牌产品，甚至还有一部分是以仿制、来件组装或“贴牌”方式生产。单从技术角度来看，国产小型 PLC 与国际知名品牌小型 PLC 差距正在缩小。例如和利时、上海新华和汇川等公司生产的微型 PLC 已经比较成熟，其可靠性在许多低端应用中得到了验证，但其知名度与世界先进水平还有相当的差距。

1.2　可编程序控制器的结构和工作原理

1.2.1　可编程序控制器的硬件组成

可编程序控制器种类繁多，但其基本结构和工作原理相同。可编程序控制器的功能结构区由 CPU（中央处理器）、存储器和输入模块/输出模块 3 大部分组成，如图 1-1 所示。

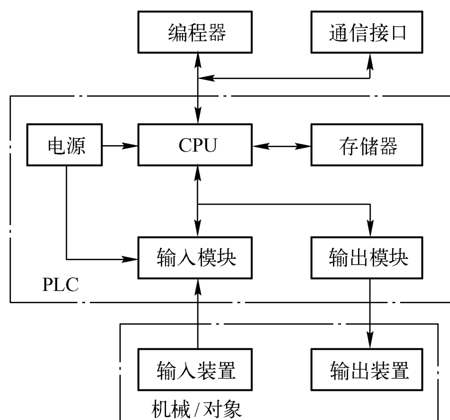


图 1-1　可编程序控制器结构框图

1．CPU（中央处理器）

CPU 的功能是完成 PLC 内所有的控制和监视操作。中央处理器一般由控制器、运算器和寄存器组成。CPU 通过数据总线、地址总线和控制总线与存储器、输入/输出接口电路连接。

2．存储器

在 PLC 中使用两种类型的存储器：一种是只读类型的存储器，如 EPROM 和 EEPROM，另一种是可读/写的随机存储器 RAM。PLC 的存储器分为 5 个区域，如图 1-2 所示。

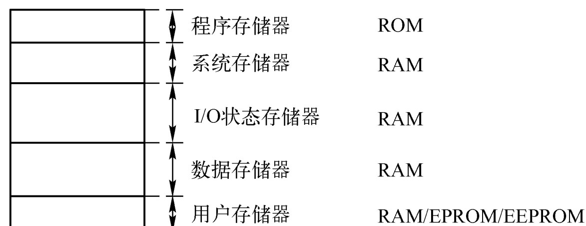


图 1-2　存储器的区域划分

程序存储器的类型是只读存储器（ROM），PLC 的操作系统存放在这里，程序由制造商固化，通常不能修改。存储器中的程序负责解释和编译用户编写的程序、监控 I/O 接口的状态、对 PLC 进行自诊断、扫描 PLC 中的程序等。系统存储器属于随机存储器（RAM），主要用于存储中间计算结果和数据及系统管理。有的 PLC 厂家用系统存储器存储一些系统信息，如错误代码等。系统存储器不对用户开放。I/O 状态存储器属于随机存储器，用于存储 I/O 装置的状态信息，每个输入模块和输出模块都在 I/O 映像表中分配一个地址，而且这个地址是唯一的。数据存储器属于随机存储器，主要用于数据处理，为计数器、定时器、算术计算和过程参数提供数据存储。有的厂家将数据存储器细分为固定数据存储器和可变数据存储器。用户编程存储器的类型可以是随机存储器、可擦除存储器（EPROM）和电擦除存储器（EEPROM），高档的 PLC 还可以用 FLASH。用户编程存储器主要用于存放用户编写的程序。存储器的关系如图 1−3 所示。

只读存储器可以用来存放系统程序，PLC 断电后再上电，系统内容不变且重新执行。只读存储器也可用来固化用户程序和一些重要参数，以免因偶然操作失误而造成程序和数据的破坏或丢失。随机存储器中一般存放用户程序和系统参数。当 PLC 处于编程工作状态时，CPU 从 RAM 中取指令并执行。用户程序执行过程中产生的中间结果也在 RAM 中暂时存放。RAM 通常由 CMOS 型集成电路组成，功耗小，但断电时内容消失，所以一般使用大电容或后备锂电池保证掉电后 PLC 的内容在一定时间内不丢失。

3．输入/输出接口

可编程序控制器的输入和输出信号可以是开关量或模拟量。输入/输出接口是 PLC 内部弱电（low power）信号和工业现场强电（high power）信号联系的桥梁。输入/输出接口主要有两个作用：一是利用内部的电隔离电路将工业现场和 PLC 内部进行隔离，起保护作用；二是调理信号，把不同的信号（如强电、弱电信号）调理成 CPU 可以处理的信号（5V、3.3V 或 2.7V 等），如图 1−4 所示。

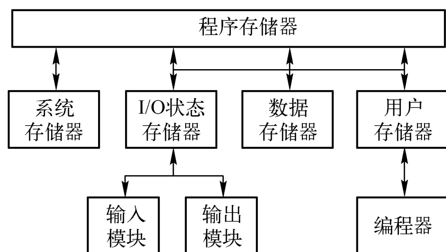


图 1−3　存储器的关系

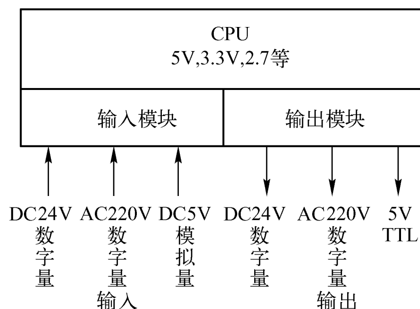


图 1−4　输入/输出接口

输入/输出接口模块是 PLC 系统中最大的部分，输入/输出接口模块通常需要电源，输入电路的电源可以由外部提供，对于模块化的 PLC 还需要背板（安装机架）。

（1）输入接口电路

1）输入接口电路的组成和作用。输入接口电路由接线端子、信号调理和转换电路、状态显示电路、电隔离电路和多路开关模块组成，如图 1−5 所示。现场的信号必须连接在输入端子才可能将信号输入到 CPU 中，它提供了外部信号输入的物理接口；信号调理和转换电



路十分重要，可以将工业现场的信号（如强电 AC220V 信号）转化成电信号（CPU 可以识别的弱电信号）；电隔离电路主要利用电隔离器件将工业现场的机械或者电输入信号和 PLC 的 CPU 的信号隔开，它能确保过高的电干扰信号和浪涌不串入 PLC 的微处理器，起保护作用。一般有 3 种隔离方式，用得最多的是光电隔离，其次是变压器隔离和干簧继电器隔离。当外部有信号输入时，输入模块上有指示灯显示，这个电路比较简单，当线路中有故障时，它帮助用户查找故障，由于氖灯或 LED 灯的寿命比较长，所以这个灯通常是氖灯或 LED 灯。多路开关模块接收调理完成的输入信号，并存储在多路开关模块中，当输入循环扫描时，多路开关模块中信号输送到 I/O 状态寄存器中。

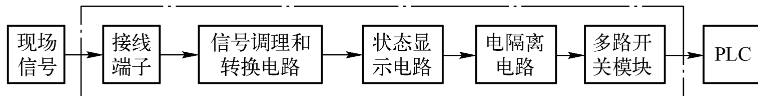


图 1-5　输入接口的结构

2）输入信号的设备的种类。输入信号可以是离散信号和模拟信号。当输入端是离散信号时，输入端的设备类型可以是限位开关、按钮、压力继电器、继电器触点、接近开关、选择开关、光电开关等，如图 1-6 所示。当输入为模拟量输入时，输入设备的类型可以是压力传感器、温度传感器、流量传感器、电压传感器、电流传感器、力传感器等。

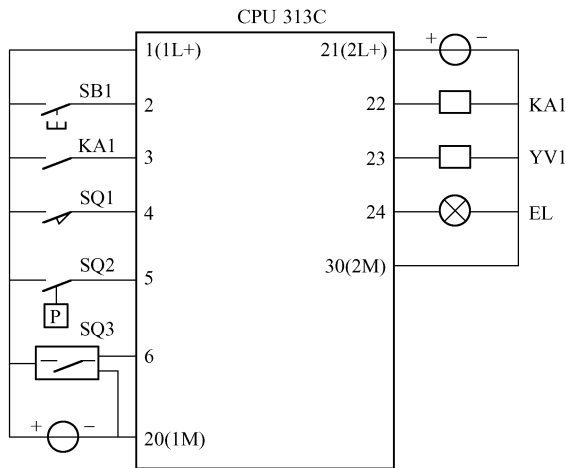


图 1-6　输入/输出接口

（2）输出接口电路

1）输出接口电路的组成和作用。输出接口电路由多路开关模块、信号锁存器、电隔离电路、状态显示电路、输出转换电路和接线端子组成，如图 1-7 所示。在输出扫描期间，多路开关模块接收来自映像表中的输出信号，并对这个信号的状态和目标地址进行译码，最后将信息送给信号锁存器；信号锁存器是将多路开关模块的信号保存起来，直到下一次更新；输出接口的电隔离电路作用和输入模块的一样，但是由于输出模块输出的信号比输入信号要强得多，因此要求隔离电磁干扰和浪涌的能力更高；输出转换电路将隔离电路送来的信号放大成足够驱动现场设备的信号，放大器件可以是双向晶闸管、三极管和干簧继电器等；输出的接线端子用于将输出模块与现场设备相连接。

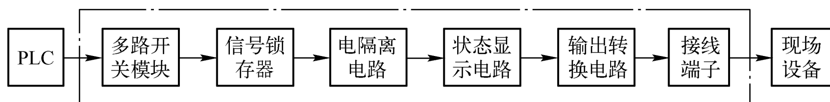


图 1–7　输出接口的结构

可编程序控制器有 3 种输出接口形式，继电器输出、晶体管输出和晶闸管输出形式。继电器输出形式的 PLC 的负载电源可以是直流电源或交流电源，但其输出频率较低。晶体管输出的 PLC 负载电源是直流电源，其输出频率较高。晶闸管输出形式的 PLC 的负载电源是交流电源。选型时要特别注意 PLC 的输出形式。

2）输出信号的设备的种类。输出信号可以是离散信号和模拟信号。当输出端是离散信号时，输出端的设备类型可以是电磁阀的线圈、电动机起动器、控制柜的指示器、接触器线圈、LED 等、指示灯、继电器线圈、报警器和蜂鸣器等，如图 1–6 所示。当输出为模拟量输出时，输出设备的类型可以是流量阀、AC 驱动器（如交流伺服驱动器）、DC 驱动器、模拟量仪表、温度控制器和流量控制器等。

1.2.2　可编程序控制器的工作原理

PLC 是一种存储程序的控制器。用户根据某一对象的具体控制要求，编制好控制程序后，用编程器将程序输入到 PLC（或用计算机下载到 PLC）的用户程序存储器中寄存。PLC 的控制功能就是通过运行用户程序来实现的。

PLC 运行程序的方式与微型计算机相比有较大的不同。微型计算机运行程序时，一旦执行到 END 指令，程序运行结束；而 PLC 从 0 号存储地址所存放的第一条用户程序开始，在无中断或跳转的情况下，按存储地址号递增的方向顺序逐条执行用户程序，直到 END 指令结束，然后再从头开始执行，并周而复始地重复，直到停机或从运行（RUN）切换到停止（STOP）工作状态。这种执行程序的方式称为扫描工作方式。每扫描完一次程序就构成一个扫描周期。另外，PLC 对输入、输出信号的处理与微型计算机不同。微型计算机对输入、输出信号实时处理，而 PLC 对输入、输出信号是集中批处理。下面具体介绍 PLC 的扫描工作过程。其运行和信号处理示意如图 1–8 所示。

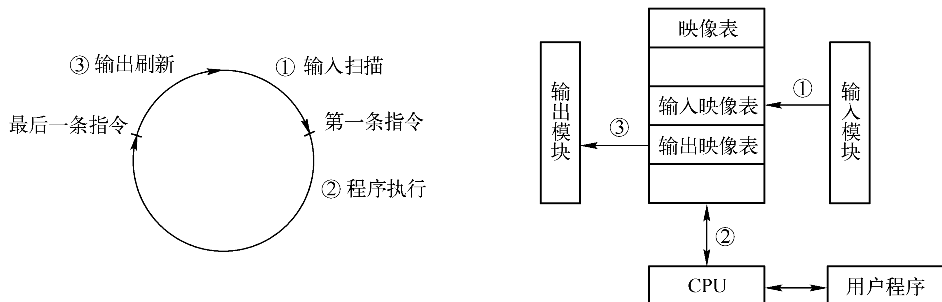


图 1–8　PLC 内部运行和信号处理示意图

PLC 扫描工作方式主要分为 3 个阶段：输入扫描、程序执行、输出刷新。

1．输入扫描

PLC 在开始执行程序之前，首先扫描输入端子，按顺序将所有输入信号读入到寄存器–输入状态的输入映像寄存器中，这个过程称为输入扫描。PLC 在运行程序时，所需的输入信



号不是取输入端子上的信息，而是取输入映像寄存器中的信息。在本工作周期内这个采样结果的内容不会改变，只有到下一个扫描周期输入扫描阶段才被刷新。PLC 的扫描速度很快，取决于 CPU 的时钟速度。

2．程序执行

PLC 完成了输入扫描工作后，按顺序从 0 号地址开始的程序进行逐条扫描执行，并分别从输入映像寄存器、输出映像寄存器以及辅助继电器中获得所需的数据进行运算处理。再将程序执行的结果写入输出映像寄存器中保存。但这个结果在全部程序未被执行完毕之前不会送到输出端子上，也就是物理输出是不会改变的。其扫描时间取决于程序的长度、复杂程度和 CPU 的功能。

3．输出刷新

在执行到 END 指令，即执行完用户所有程序后，PLC 上将输出映像寄存器中的内容送到输出锁存器中进行输出，驱动用户设备。扫描时间取决于输出模块的数量。

从以上的介绍可以知道，PLC 程序扫描特性决定了 PLC 的输入和输出状态并不能在扫描的同时改变，例如，一个按钮开关的输入信号的输入刚好在输入扫描之后，那么这个信号只有在下一个扫描周期才能被读入。

上述 3 个步骤是 PLC 的软件处理过程，可以认为就是程序扫描时间。扫描时间通常由 3 个因素决定：一是 CPU 的时钟速度，越高档的 CPU，时钟速度越快，扫描时间越短；二是 I/O 模块的数量，模块数量越少，扫描时间越短；三是程序的长度，程序长度越短，扫描时间越短。一般的 PLC 执行容量为 1KB 的程序约需要的扫描时间是 1～10ms。

1.2.3　可编程序控制器的立即输入、输出功能

比较高档的 PLC 都有立即输入、输出功能。

1．立即输出功能

所谓立即输出功能就是输出模块在处理用户程序时，能立即被刷新。PLC 临时挂起（中断）正常运行的程序，将输出映像表中的信息输送到输出模块，立即进行输出刷新，然后再回到程序中继续运行。立即输出的示意图如图 1-9 所示。注意，立即输出功能并不能立即刷新所有的输出模块。

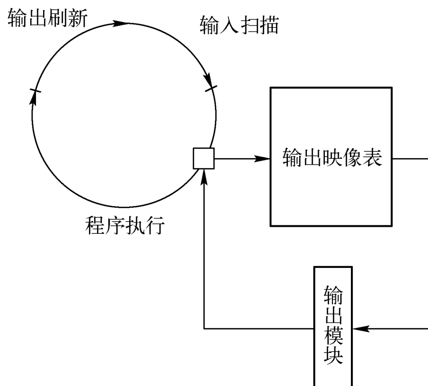


图 1-9　立即输出过程

2．立即输入功能

立即输入功能适用于要求对反应速度很严格的场合，例如几毫秒的时间对于控制来说十分关键的情况下。立即输入时，PLC 立即挂起正在执行的程序，扫描输入模块，然后更新特定的输入状态到输入映像表，最后继续执行剩余的程序。立即输入的示意图如图 1-10 所示。

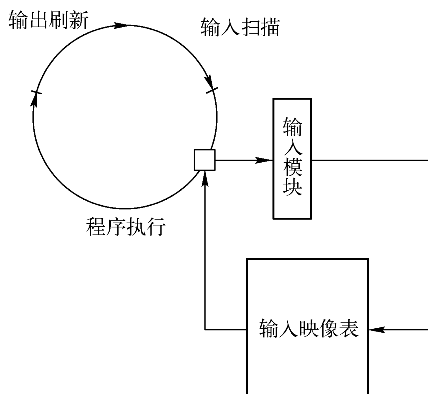


图 1-10　立即输入过程

小结

重点难点总结

1．PLC 的应用范围。

2．PLC 的工作机理和结构。

习题

1．PLC 的主要性能指标有哪些？

2．PLC 主要用在哪些场合？

3．PLC 是怎样分类的？

4．PLC 的发展趋势是什么？

5．PLC 的结构主要有哪几个部分组成？

6．PLC 的输入和输出模块主要有哪几个部分组成？每部分的作用是什么？

7．PLC 的存储器可以细分为哪几个部分？

8．PLC 是怎样进行工作的？

9．举例说明常见的哪些设备可以作为 PLC 的输入设备和输出设备？

10．什么是立即输入和立即输出？在何种场合应用？

11．S7 系列的 PLC 有哪几类？

12．PLC 控制与继电器控制有何优缺点？

第 2 章

S7-300/400 PLC的硬件介绍

本章介绍常用 S7-300/400 PLC 的 CPU 模块、数字量输入/输出模块、模拟量输入/输出模块、功能模块、机架和电源模块的功能、接线和安装，这些内容是后续程序设计和控制系统设计所必备的。

2.1　S7-300 PLC常用模块及其接线

2.1.1　西门子PLC简介

德国的西门子（SIEMENS）公司是欧洲最大的电子和电气设备制造商之一，生产的 SIMATIC 可编程序控制器在欧洲处于领先地位。其第一代可编程序控制器是 1975 年投放市场的 SIMATIC S3 系列控制系统。之后在 1979 年，西门子公司将微处理器技术应用到可编程序控制器中，研制出了 SIMATIC S5 系列，取代了 S3 系列，目前 S5 系列产品仍然在工业现场使用，在 20 世纪末，西门子又在 S5 系列的基础上推出了 S7 系列产品。最新的 SIMATIC 产品为 SIMATIC S7 和 C7 等几大系列。C7 是基于 S7-300 PLC 性能，同时集成了 HMI。

SIMATIC S7 系列产品分为：通用逻辑模块（LOGO!）、S7-200 PLC、S7-1200 PLC、S7-300 PLC 和 S7-400 PLC 5 个产品系列。S7-200 PLC 是在德州仪器公司的小型 PLC 的基础上发展而来，因此其指令系统、程序结构、编程软件，这些和 S7-300/400 PLC 有较大的区别。S7-1200 PLC 是在 2009 年才推出的新型小型 PLC，定位于 S7-200 PLC 和 S7-300 PCL 产品之间。S7-300/400 PLC 是由西门子的 S5 系列发展而来。西门子的 PLC 产品系列的定位见表 2−1。

表 2−1　SIMATIC 控制器的定位

序号	控制器	定位	主要任务和性能特征
1	LOGO！	低端独立自动化系统中简单的开关量解决方案和智能逻辑控制器	简单自动化 作为时间继电器、计数器和辅助接触器的替代开关设备 模块化设计，柔性应用 有数字量、模拟量和通信模块 用户界面友好，配置简单 使用拖放功能的智能电路图开发
2	S7-200	低端的离散自动化系统和独立自动化系统中使用的紧凑型逻辑控制器模块	串行模块结构、模块化扩展 紧凑设计，CPU 集成 I/O 实时处理能力，高速计数器、报警输入、中断 易学易用的软件 多种通信选项

（续）

序号	控制器	定位	主要任务和性能特征
3	S7-1200	低端的离散自动化系统和独立自动化系统中使用的小型控制器模块	可升级及灵活的设计 集成了 PROFINET 接口 集成了强大的计数、测量、闭环控制及运动控制功能 直观高效的 STEP 7 Basic 工程系统可以直接组态控制器和 HMI
4	S7-300	中端的离散自动化系统中使用的控制器模块	通用型应用和丰富的 CPU 模块种类 高性能 模块化设计，紧凑设计 由于使用 MMC 存储程序和数据，系统免维护
5	S7-400	高端的离散和过程自动化系统中使用的控制器模块	特别高的通信和处理能力 定点加法或乘法的指令执行速度最快为 0.03μs 大型 I/O 框架和最高 20MB 的主内存 快速响应，实时性强，垂直集成 支持热插拔和在线 I/O 配置，避免重启 具备等时模式，可以通过 PROFIBUS 控制高速机器

2.1.2　S7-300 PLC常用模块及其接线

1．S7-300 PLC的基本结构

S7-300 PLC 是模块化结构设计的 PLC，各个单独模块之间可进行广泛组合和扩展。它的主要组成部分有电源模块（PS）、中央处理器模块（CPU）、导轨（RACK）、接口模块（IM）、信号模块（SM）和功能模块（FM）等。S7-300 PLC 可以通过 MPI 接口直接与编程器（PG）、操作面板（OP）和其他 S7 系列 PLC 相连。其实物如图 2−1 所示，其系统构成如图 2−2 所示。

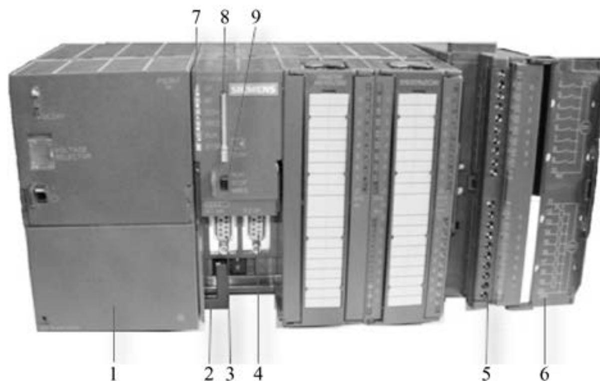


图 2−1　S7-300 PLC 实物图

1—电源模块　2—24V DC 连接器　3—MPI 接口　4—DP 接口

5—前连接器　6—前盖　7—状态和故障指示灯　8—存储卡　9—模式开关

电源模块（PS）用于向 CPU 及其扩展模块提供 DC24V 电源。电源模块与 CPU 之间用电缆（连接器）连接，而不是用背板供电，因此有的设计者为了省钱用普通开关电源取代西门子的电源模块，但这样做是不被西门子公司推荐的。目前电源模块有 2A、5A 和 10A 3 种规格，可根据实际情况选用。

中央处理器模块（CPU），S7-300 PLC 的 CPU 模块主要包括 CPU 312、CPU 312C、CPU 313C、CPU 313C-PtP、CPU 314-2DP 等型号。有的型号还有不同的版本号（如 CPU 314-2DP 目前有 2.0 版和 2.6 版），每种 CPU 有其不同的性能。CPU 型号中的“C”表示紧凑



型 CPU；CPU 型号中的“DP”表示带有 9 针的 DP 接口；CPU 型号中的“PtP”表示带有 15 针的 PtP 接口；有的 CPU 上还带有输入/输出端子。

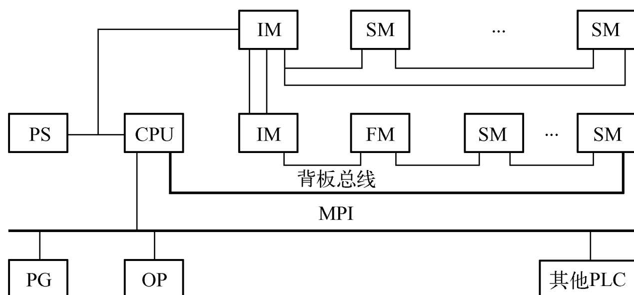


图 2-2　S7-300 PLC 系统构成图

导轨（RACK）是安装 S7-300 PLC 各类模块的机架，它是特制的异形板，其标准长度有 160mm、482mm、530mm、830mm 和 2000mm，可以根据实际选用，此导轨可以切割成实际需要的长度。

信号模块（SM）是数字量 I/O 模块和模拟量 I/O 模块的总称。信号模块主要有 SM321（数字量输入）、SM322（数字量输出）、SM331（模拟量输入）和 SM332（模拟量输出）等模块。每个模块都带有一个总线连接器，用于 CPU 和其他模块之间的数据通信。

功能模块（FM）主要用于对实时性和存储量要求高的控制任务。如计数模块 FM350、定位模块 FM353 等。

通信处理模块（CP）用于 PLC 之间、PLC 与计算机和其他智能设备之间的通信，可以将 PLC 接入工业以太网、PROFIBUS 和 AS-I 网络，或用于串行通信。它可以减轻 CPU 处理通信的负担，并减少用户对通信功能的编程工作。

接口模块（IM）用于多机架配置时连接主机架（CR）和扩展机架（ER）。S7-300 PLC 通过分布式的主机架和连接的扩展机架（最多可连接 3 个扩展机架），可以最多操作 32 个模块。

2．S7-300 PLC的CPU模块

S7-300 PLC 的 CPU 模块共有 20 多个不同的型号，按照性能等级划分，可涵盖各种应用领域。主要分以下几类：

（1）CPU 模块的分类

① 紧凑型 CPU。包括 CPU 312C、313C、313C-PtP、313C-2DP、314C-PtP 和 314C-2DP。各 CPU 均有计数、频率测量和脉冲宽度调制功能。有的带有定位功能，有的带有 I/O。

② 标准型 CPU。包括 CPU 312、313、314、315、315-2DP 和 316-2DP。

③ 户外型 CPU。包括 CPU 312 IFM、314 IFM、314 户外型和 315-2DP。这种模型在恶劣的环境下使用。

④ 高端 CPU。包括 CPU 317-2DP 和 CPU 318-2DP。具有大容量的程序存储器和 PROFIBUS-DP 接口，可以用于大规模的 I/O 配置（如 CPU 318-2DP 最多可以配置 65536 个

数字 I/O），建立分布式 I/O 结构。

⑤ 故障安全型 CPU。CPU 315F，不需要对故障 I/O 进行额外接线，可以组态成一个故障安全型自动化系统。

（2）CPU 的状态与故障显示 LED

CPU 317-2DP 的面板如图 2-3 所示，其他的 CPU 的面板和 CPU 317-2DP 类似。

SF（系统出错/故障显示，红色）：CPU 硬件故障或软件错误时亮。

BATF（电池故障，红色）：电池电压低或没有电池时亮。

DC 5V（＋5V 电源指示，绿色）：5V 电源正常时亮。

FRCE（强制，黄色）：至少有一个 I/O 被强制时亮。

RUN（运行方式，绿色）：CPU 处于 RUN 状态时亮；重新启动时以 2Hz 的频率闪亮；HOLD（单步、断点）状态时以 0.5Hz 的频率闪亮。

STOP（停止方式，黄色）：CPU 处于 STOP、HOLD 状态或重新启动时常亮。

BUSF（总线错误，红色）。

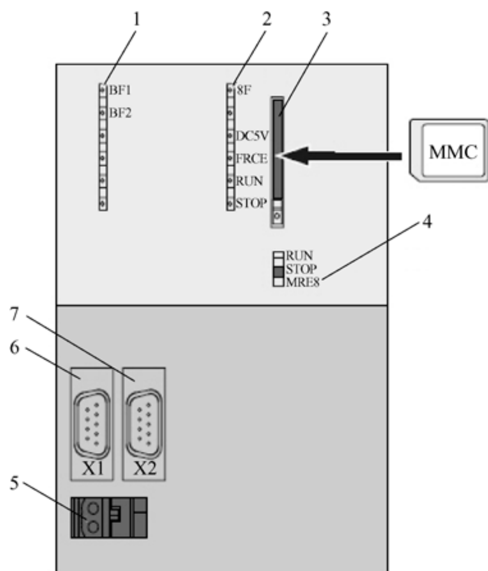


图 2-3　CPU 317-2DP 的面板

1—总线故障指示器　2—状态和错误显示　3—微存储卡（MMC）的插槽

4—模式选择器开关　5—电源连接　6—接口 X1 （MPI/DP）　7—接口 X2 （DP）

（3）模式选择开关

RUN 模式：CPU 执行用户程序。

STOP 模式：CPU 不执行用户程序。

MRES：CPU 存储器复位，带有用于 CPU 存储器复位按钮功能的模式选择器开关位置。通过模式选择器开关进行 CPU 存储器复位需要特定操作顺序。

复位存储器操作：通电后从 STOP 位置扳到 MRES 位置，“STOP”LED 熄灭 1s，亮 1s，再熄灭 1s 后保持亮。放开开关，使它回到 STOP 位置，然后又回到 MRES，“STOP”



LED 以 2Hz 的频率至少闪动 3s，表示正在执行复位，最后“STOP”LED 一直亮。

（4）紧凑型 CPU 的接线

紧凑型 CPU 的接线基本类似，因此以 CPU 314C-2DP 为例讲解紧凑型 CPU 的接线。

1）电源模块和CPU的接线。电源模块和CPU 接线比较简单，S7-300 PLC的接线都相同。先打开PS 307 电源模块和CPU 前面板，再打开PS 307 上的电缆夹，接着将电源电缆连接到PS 307 的L1、N和保护接地 （PE） 端。最后将PS 307 上的端子M 和L+ 连接到CPU 上的M 和L+ 端子。电源模块和CPU接线如 图 2−4 所示。

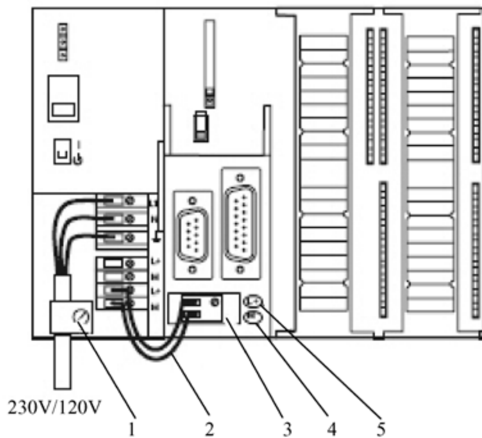


图 2−4 电源模块和CPU接线

1—电缆夹 2—连接电缆 3—可拆卸的电源连接器 4—标记“M” 5—标记“L+”

2）数字 I/O 的接线。紧凑型 CPU 一般带有数字 I/O，输入为 PNP 型输入（高电平有效），这一点不同于 S7-200 PLC，后者为 PNP 型和 NPN 型输入可选，因此输入端若要连接接近开关，通常应选择 PNP 型。输入端的 1 号端子与 DC24 V 相连，而 20 号端子与 0V 相连，10 号和 11 号端子不使用，其余输入端的端子都可以作为输入点使用。

输出也是 PNP 输出（高电平有效），这一点与 S7-200 PLC 相同。输出端的 21 号和 31 号端子与 DC24 V 相连，而 30 号和 40 号端子与 0V 相连，其余输出端的端子都可以作为输出点使用。数字 I/O 的接线如图 2−5 所示。S7-300 PLC 的接线图一般印刷在机壳上，当然也可从西门子公司的网站下载。

【例 2-1】 某设备的控制器为 CPU 314C-2DP，控制三相交流电动机的起停控制，并有一只接近开关限位，请设计接线图。

【解】

根据题意，只需要 3 个输入点和一个输出点。因此使用 CPU 314C-2DP 上集成的 I/O 即可，输入端和输出端都是 PNP 型，因此接近开关只能用 PNP 型的接近开关（不用转换电路时），接线图如图 2−6 所示。交流电动机的起停一般要使用交流接触器，交流回路由读者自行设计，在此不作赘述。

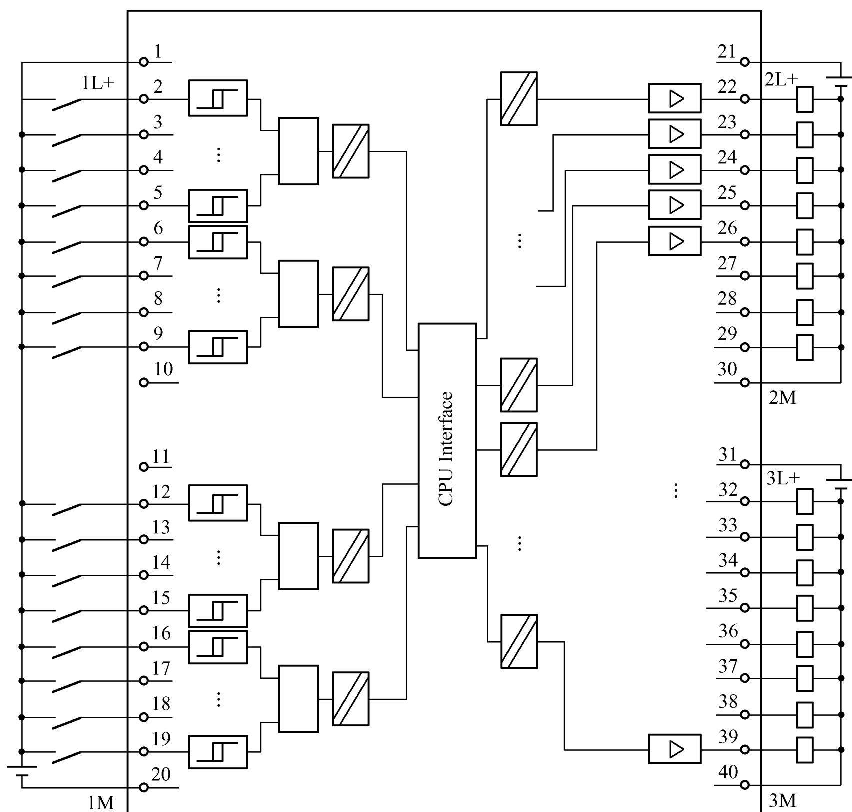


图 2-5　CPU 314C-2DP的数字I/O的接线

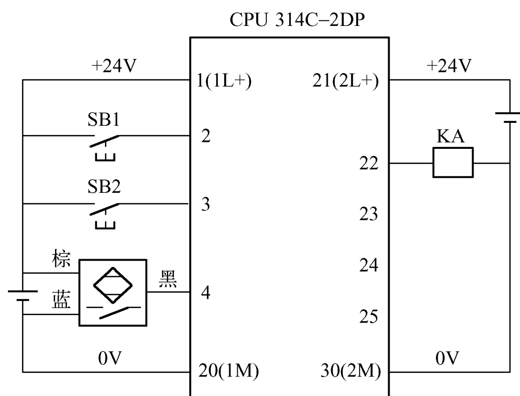


图 2-6　例 2-1 的接线图

3）模拟量 I/O 的接线。紧凑型 CPU 除了带有数字 I/O，一般还自带模拟 I/O，这无疑是非常方便的，对于要求不是很高控制系统使用紧凑型 CPU 是非常经济的。

如图 2-7 所示，CPU 314C-2DP有 5 个模拟输入通道，其中输入通道 0（CH0）、输入通道 1（CH1）、输入通道 2（CH2）和输入通道 3（CH3），每个输入通道都可以输入电压或者



电流信号，但二者只能选择其一，不能同时输入，例如在通道 0 中要采集电压信号，那么只要将电压信号连接在 2 号端子和 4 号端子之间即可，3 号端子悬空。

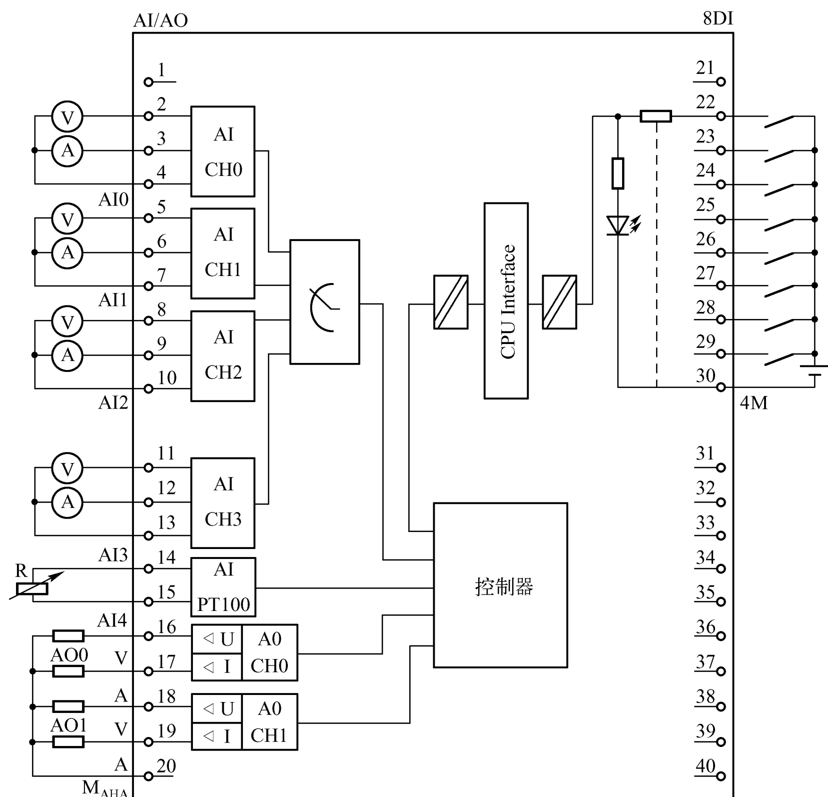


图 2-7　CPU 314C-2DP的模拟I/O接线

14 号和 15 号端子上只能接入热电阻 Pt100。

AO0 和 AO1 是两个模拟输出通道，每个模拟输出通道都可输出模拟电压信号和电流信号，但二者只能选择其一，不能同时输出，例如在 AO0 中要输出电压信号，那么只要将电压信号连接在 16 号端子和 20 号端子之间即可，17 号端子悬空。读者可能已经发现这里只谈到接线端子，却未提到输入/输出地址，有关地址相关内容将在后续章节讲解。

3．数字量模块

S7-300 PLC 有多种型号的数字量输入/输出（I/O）模块供选择。以下将介绍数字量输入模块 SM321、数字量输出模块 SM322 和数字量输入/输出模块 SM323。

（1）数字量输入模块 SM321

1）数字量输入模块 SM321 的工作原理。数字量模块用于采集现场过程的数字信号，有的模块采集直流信号，有的模块则可采集交流信号，并把它转化成 PLC 内部的信号电平。对于现场输入元件，仅要求提供开关触点即可。输入信号进入模块后，一般经过光电隔离、滤波，然后送到输入缓冲寄存器等，到 CPU 采样时，采样信号经过背板总线进入输入映像区。




用于采集直流信号的模块称为直流输入模块，一般的输入电压为 DC24V，如图 2−8 所示。用于采集交流信号的模块称为交流输入模块，一般的输入电压为 AC120V 或者 AC230V，如图 2−9 所示。

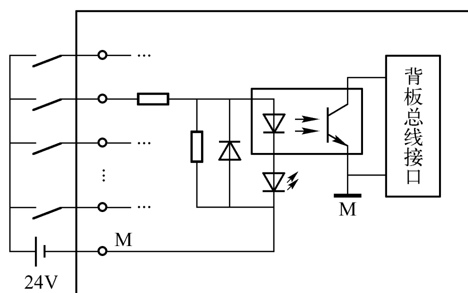


图 2−8　直流 数字量模块

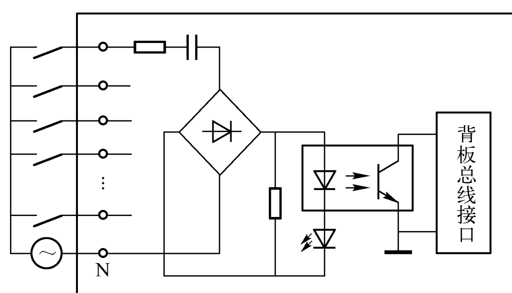


图 2−9　交流 数字量模块

2）数字量输入模块 SM321 的接线。数字量输入模块 SM321 有多种型号，以下介绍 3 种有代表性的输入模块的接线。

直流数字量输入模块多为 PNP 输入，只有个别型号为 NPN 输入，如图 2−10 所示的 SM321 模块是 16 点直流 PNP 输入模块，读者在判断是否为 PNP 型，只要判断输入有效信号是否为高电平，很明显见图 2−10 中，当开关闭合时输入的是高电平，因此是 PNP 输入。输入开关的一端与模块的端子相连，而另一端则与同一个电源的 24V 相连。

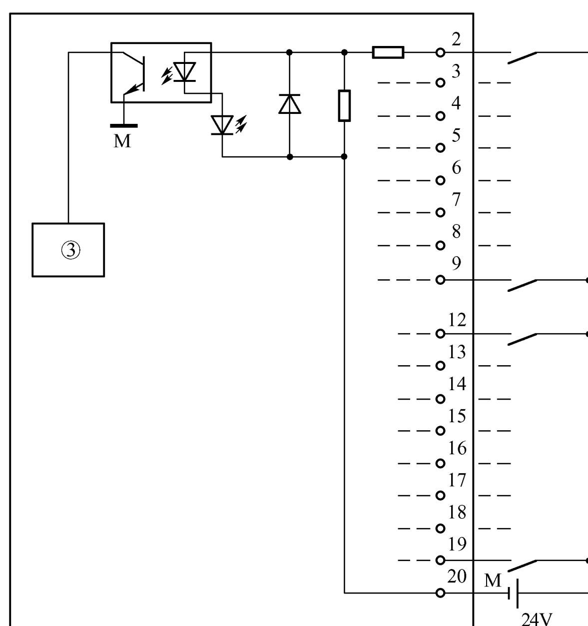


图 2−10　直流 数字量输入模块接线图（PNP）

PNP 型数字量输入模块 SM321 的接线与紧凑型 CPU 的数字量的输入类似。

如图 2−11 所示的 SM321 模块是 16 点直流 NPN 输入模块，其有效输入信号是低电平。输入开关的一端与模块的端子相连，而另一端则与同一个电源的 0V 相连。在接线时要特别注意。

【关键点】西门子的直流输入模块一般是 PNP 型输入，因此若系统使用接近开关时，最



好选用 PNP 接近开关。交流输入模块则不存在此问题，只要选用交流接近开关即可。

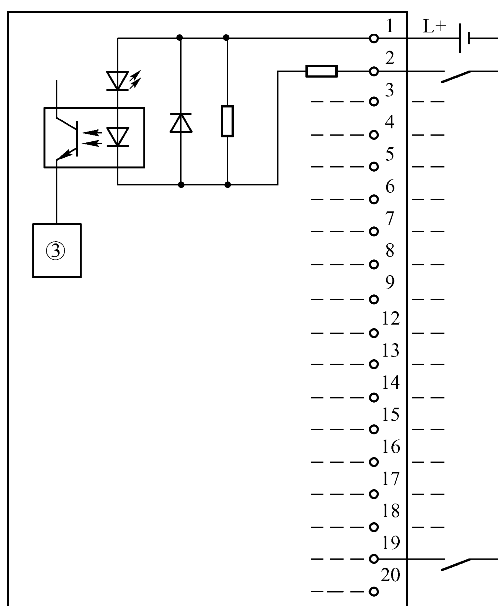


图 2-11　直流 数字量输入模块接线图（NPN）

交流 数字量输入模块的接线如图 2-12 所示。交流电源零线（或者相线）与 1N、2N、3N和 4N相连，交流电源相线或者零线与输入开关一端相连，而开关的另一端与模块的输入接线端子相连。输入开关可以是按钮、交流接近开关等。

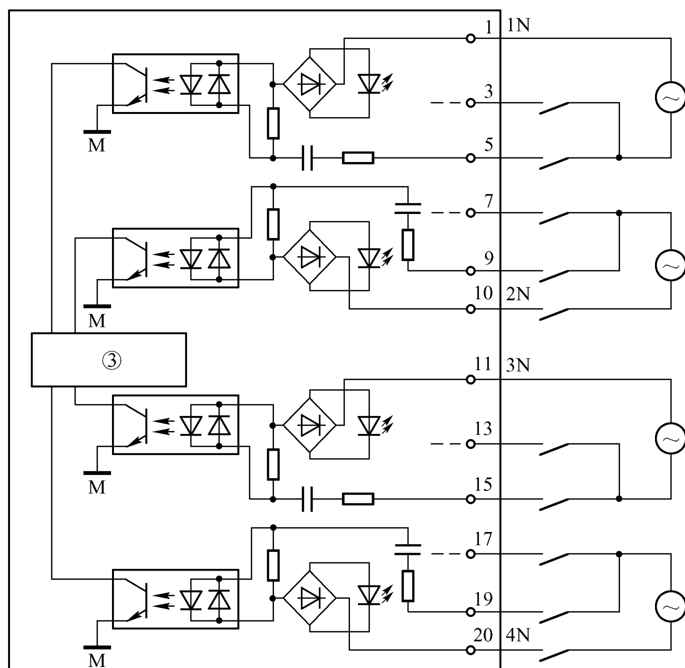


图 2-12　交流 数字量输入模块接线图

（2）数字量输出模块 SM322

1）数字量输出模块 SM322 的工作原理。数字量输出模块将 PLC 内部的电平信号转换成外部过程需要的电平信号，同时具有隔离和信号放大的作用，可直接用于驱动电磁阀、接触器、继电器和小功率器件等。

输出模块有晶体管输出（直流输出）、晶闸管输出（交流输出）和继电器输出（交直流输出）3 种类型。晶体管输出模块只能驱动直流负载，晶闸管输出模块只能驱动交流负载，而继电器输出模块可以驱动交流和直流负载。从响应速度来看，晶体管输出速度最快，继电器输出响应速度最慢，但从使用的灵活性来看，继电器输出最为灵活。3 种模块的内部结构如图 2−13、图 2−14 和图 2−15 所示。

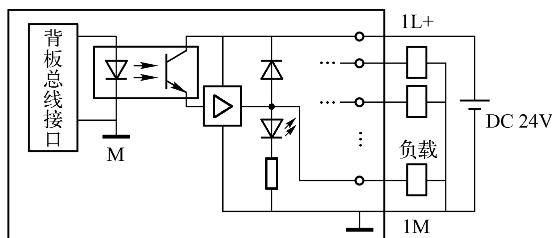


图 2−13　数字量输出模块（晶体管输出）

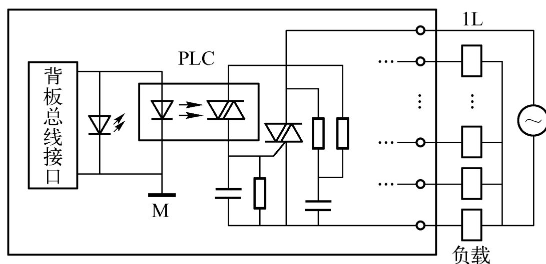


图 2−14　数字量输出模块（晶闸管输出）

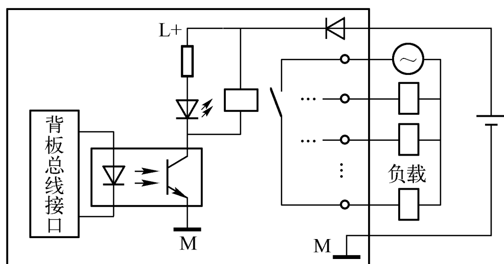


图 2−15　数字量输出模块（继电器输出）

2）数字量输出模块SM322 的接线。直流 数字量输出模块接线如 图 2−16 所示，直流 数字量输出模块是PNP输出（有效输出信号为高电平），负载的一端与模块的接线端子（图 2−16 所示 ）相连，负载的另一端与电源的 0V相连。1L+和 2L+与电源的DC24 V相连，1M和 2M与电源的 0V相连。



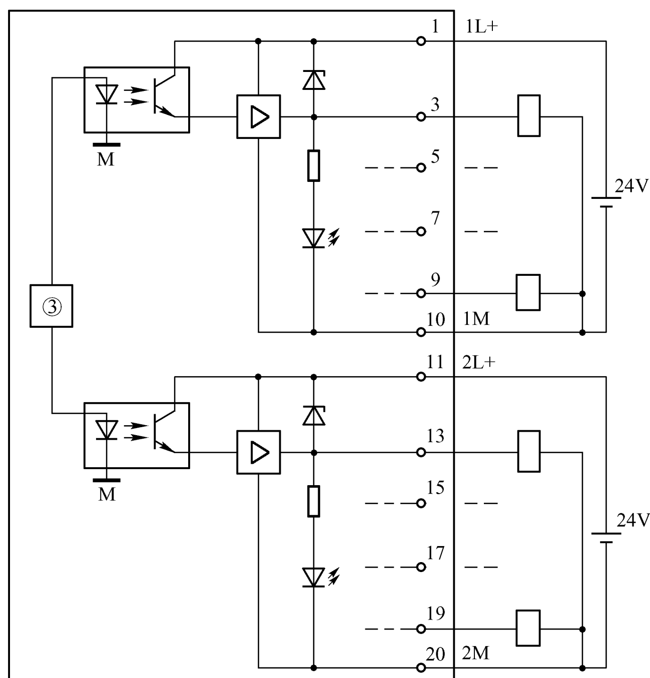


图 2-16　直流 数字量输出模块接线图

交流 数字量输出模块接线如 图 2-17 所示，交流 数字量输出模块使用相对较少，其响应速度介于 直流 数字量输出模块和 继电器 数字量输出模块之间。

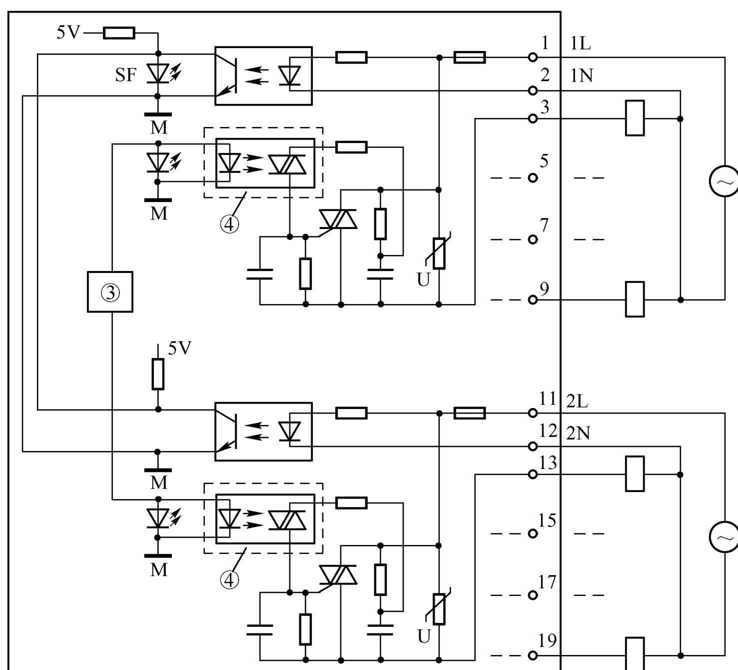


图 2-17　交流 数字量输出模块接线图

继电器 数字量输出模块接线如 图 2−18 所示 。

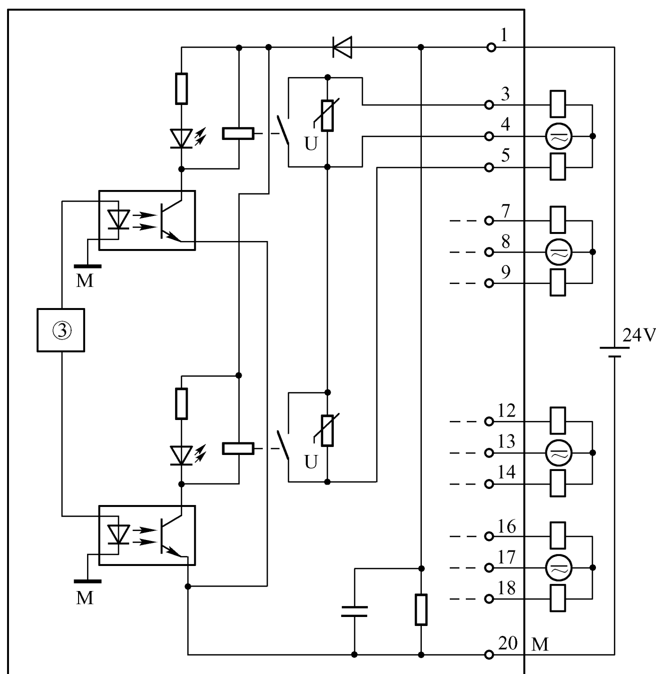


图 2−18　继电器 数字量输出模块接线图

SM323 是 S7-300 PLC 的数字量输入/输出模块，此模块兼有数字输入和数字量输出，其接线与数字输入模块和数字量输出模块类似，在此不再赘述。

4．模拟量模块

S7-300 PLC 用于模拟量输入（A/D 转换）/输出（D/A 转换）的特殊功能模块有 SM331 模拟量输入模块、SM332 模拟量输出模块、SM334 模拟量输入/输出模块和 SM335 快速模拟量输入/输出混合模块 4 类。每类中根据输入/输出通道数、分辨率、连接传感器的不同，又分为不同的规格。由于规格较多，限于篇幅，以下只介绍有代表性的模块。

（1）模拟量输入模块 SM331 连接

S7-300 PLC 单独用于模拟量输入的模块 SM331 目前有 8 种规格，8 种规格依靠订货号 6ES7 331-×××××-0AB0 中间的 5 位（×××××）进行区分。模拟量输入模块通过总线连接器与 CPU 或者扩展单元相连。

如图 2−19 所示，6ES7 331-7HF01-0AB0 模块是 8 通道 14 位 A/D 转换模块，简称 AI8×14。使用时要注意如下事项：

1）为了提高模拟量输入的可靠性，减少干扰，所有的连线均应使用屏蔽双绞线，屏蔽层要“双端接地”，接线长度要小于 200m。

2）为了缩短扫描时间，在模块的硬件配置设定时，应将模块中全部没有使用的通道设定为“禁止”状态。



3）当选择 1～5V 模拟电压输入时，如通道组只用 1 个通道，应将未使用的与同组已经使用的输入进行并联。

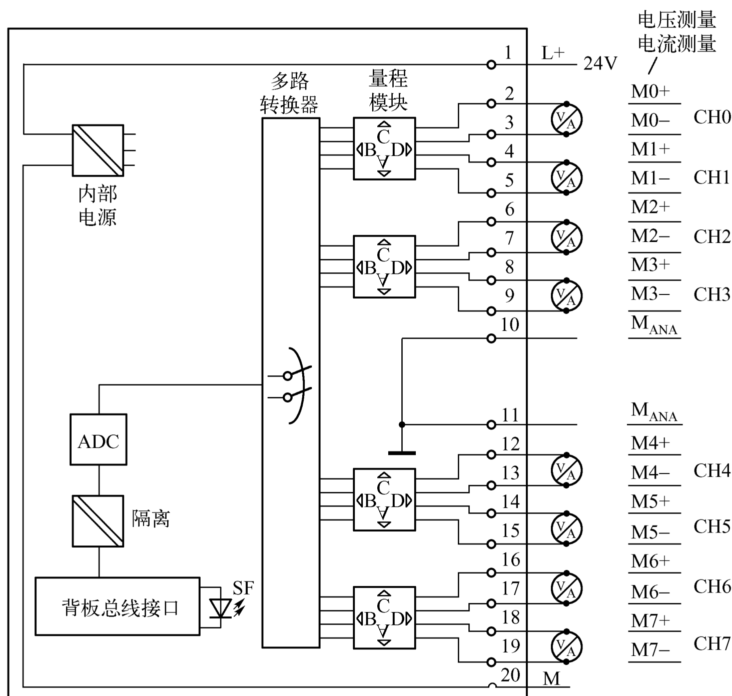


图 2−19　模拟量输入模块（6ES7 331-7HF01-0AB0）接线图

4）当选择－1～1V 模拟电压输入时，应将未使用的 Mn+与 Mn−短接，并连接到参考电位 M_{ANA}（接线端子 11）上。

5）对于两线式模拟电流输入，应将没有使用的输入端开路（不能使用模块的诊断功能），或者在输入端接 1.5~3.3kΩ的电阻（能使用模块的诊断功能）。

6）对于四线式模拟电流输入，如通道组只使用 1 个通道，应将未使用的与已经使用的输入进行串联。

7）此模块可以连接模拟电压和电流信号输入，但不能连接热电阻或者热电偶。

8）该模块只有一个 ADC，所以一次只能转换一个通道的数据，其他通道需要多路开关的切换依次进行，转换后的数据存储到各自的地址中。

如图 2−20 所示，6ES7 331-7PF01-0AB0 模块是 8 通道热电阻输入模块，简称 AI8×RTD。使用时要注意如下事项：

1）此模块只能用于热电阻输入，4 个通道使用了 4 个 ADC，一次可以完成 4 通道 A/D 转换。

2）遵守 6ES7 331-7HF01-0AB0 模块的 1）～6）项。

3）未使用的通道应连接标准电阻作为“模拟”输入信号。

4）IC0+和 IC0-（还有 IC1+和 IC2−～IC7+和 IC7−）是模块提供的电流源，不可以短接，否则会短路。

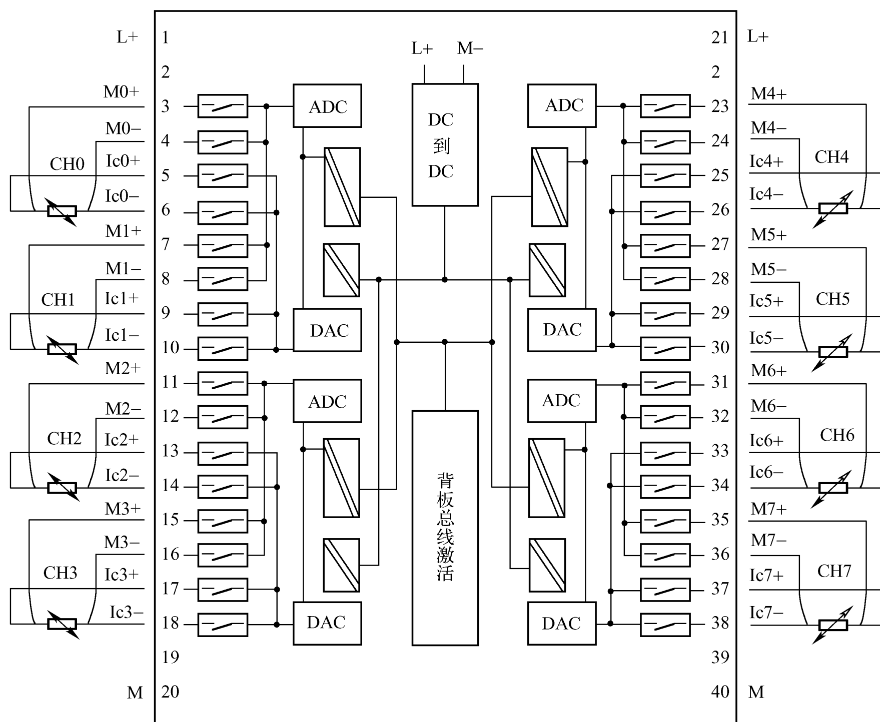


图 2-20　模拟量输入模块（6ES7 331-7PF01-0AB0）接线图

如图 2-21 所示，6ES7 331-7PF11-0AB0 模块是 8 通道热电偶输入模块，简称 AI8×TC。使用时要注意如下事项：

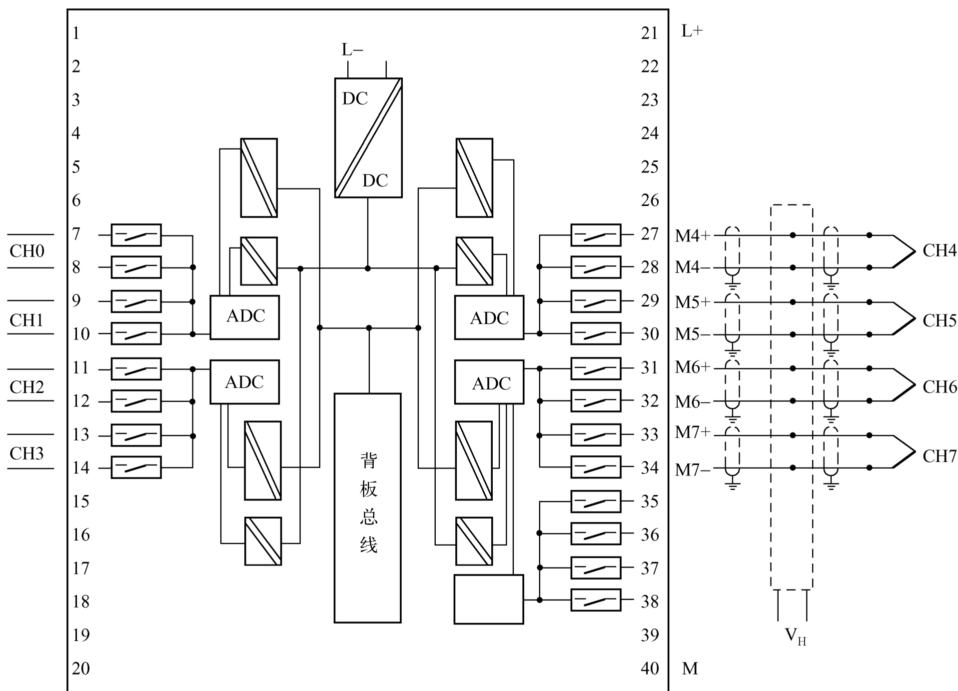


图 2-21　模拟量输入模块（6ES7 331-7PF11-0AB0）接线图



1）此模块只能用于热电偶输入，4 个通道使用了 4 个 ADC，一次可以完成 4 通道 A/D 转换。

2）遵守 6ES7331-7HF01-0AB0 模块的 1）～6）项。

3）未使用的通道应将输入端“＋”和输入端“－”短接。

（2）模拟量输出模块 SM332 连接

S7-300 PLC 单独用于模拟量输出的模块 SM332 目前有 4 种规格，分辨率为 12 位和 16 位，模拟输出通道有 2、4、8 共 3 种。模拟量输入模块通过总线连接器与 CPU 或者扩展单元相连，数据通过内部总线传送。

如图 2−22 所示，6ES7 332-5HD01-0AB0 模块是 8 通道模拟量输出模块。使用时要注意如下事项：

1）当模拟量输出为电压信号时，可采用四线式和两线式两种连接方式。当采用四线式连接时，应将负载连接到Q_{Vn}与M_{ANA}端，S+和S-用于连接负载检测端，采用这种形式可以提高模拟量输出精度。当采用两线式连接时，S+和S-开路。

2）当采用电流输出时，应将负载连接到Q_{Vn}与M_{ANA}端，S+和S-开路。

3）为了提高模拟量输入的可靠性，减少干扰，所有的连线均应使用屏蔽双绞线，对于四线式连接时，Q_{Vn}/S+与M_{ANA}/S-分别使用一对双绞线。

4）模拟量输出的最大连接距离为 200m。

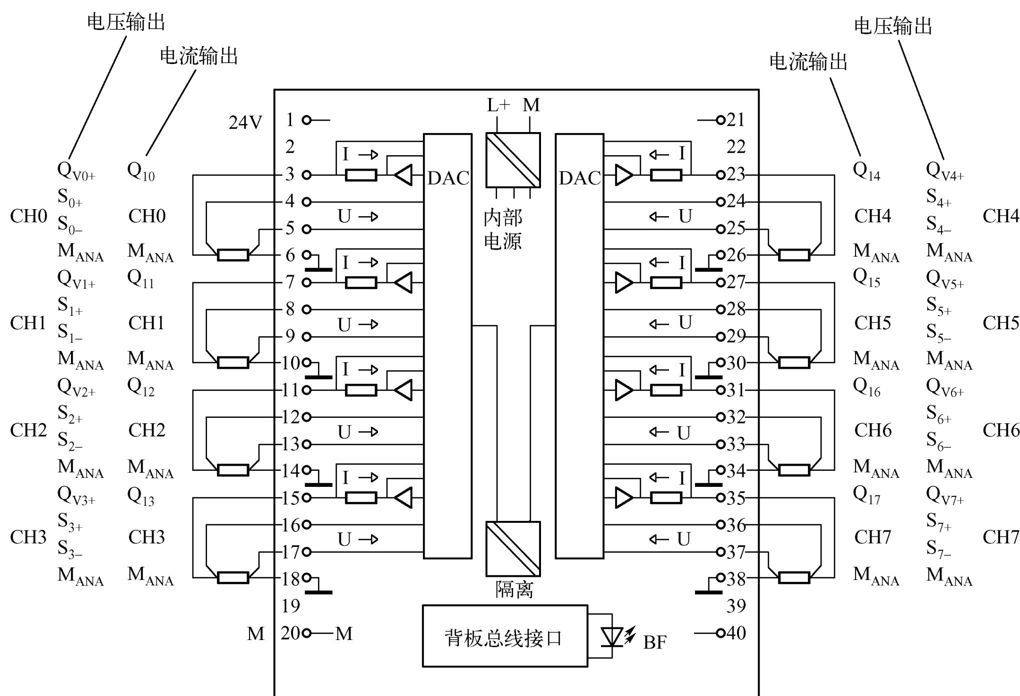


图 2−22 模拟量输出模块（6ES7 332-5HD01-0AB0）接线图

（3）模拟量输入/输出模块 SM334 连接

S7-300 PLC 同时用于模拟量输入和模拟量输出的模块是 SM334 和 SM335。SM334 是普通的 A/D、D/A 模块，目前有两种规格。SM335 是高速 A/D、D/A 转换模块，目前只有一个

规格。模拟量输入/输出模块通过总线连接器与 CPU 或者扩展单元相连，数据通过内部总线传送。

如图 2−23 所示，6ES7 334-0CE01-0AA0 模块是 4 通道模拟量输入和 2 通道模拟量输出模块。它实际上就是模拟量输入和模拟量输出模块的合并，因此其使用时的注意事项与模拟量输入和模拟量输出模块相同，在此不再赘述。

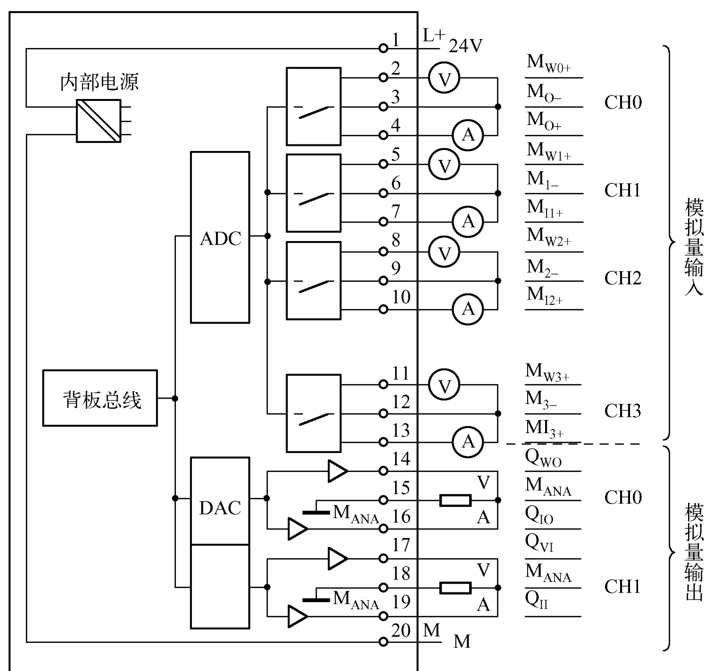


图 2−23　模拟量输入/输出模块（6ES7 334-0CE01-0AA0）接线图

SM334 模块的地址分配见表 2−2。

表 2−2　SM334 模块的地址

通道	地址
输入通道 0	模块的起始地址
输入通道 1	模块的起始地址＋2B 的地址偏移量
输入通道 2	模块的起始地址＋4B 的地址偏移量
输入通道 3	模块的起始地址＋6B 的地址偏移量
输出通道 0	模块的起始地址
输出通道 1	模块的起始地址＋2B 的地址偏移量

【例 2−2】某设备的控制器为 CPU 315-2DP，有 3 个按钮输入，数字量输入模块为 6ES7 321-1BH02-0AA0，有 4 个电磁阀（2 只为 DC24 V，2 只为 AC220 V），数字量输出模块为 6ES7 322-1HH01-0AA0，3 路电流模拟量输入，模拟量输入模块为 6ES7 331-7HF01-0AB0，3 路电压模拟量输出，模拟量输出模块为 6ES7 332-5HD01-0AB0，请设计接线图。

【解】

CPU 和扩展模块以及扩展模块之间通过总线连接器连接；数字量输入模块是 PNP 输入；数字量输出模块是继电器输出，由于电磁阀的额定电压有 DC24 V 和 AC220 V 两种，所



以一般选用继电器输出模块，而且不同额定电压的电磁阀不连接到一组；模拟量输出信号为电压信号，采用 4 线式连接（当然也可以采用 2 线式连接），其接线图如图 2-24 所示。

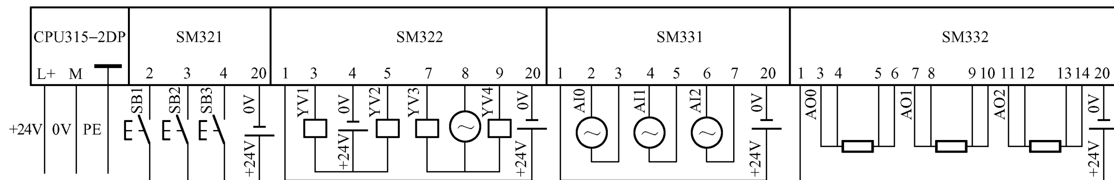


图 2-24　例 2-2 的接线图

5．S7-300 PLC的通信处理模块

通信处理模块为 PLC 接入 PROFIBUS 现场总线、工业以太网和串行通信等提供了极大的便利。通过集成在 STEP 7 中的参数化处理工具可进行简便的参数设置。

CP340 通过点对点连接进行串行通信的经济型解决方案。 它具有 3 种物理接口，分别是 RS-232C（V.24）、20 mA（TTY）和 RS 422/RS 485（X.27）。CP340 执行的协议有 ASCII 、3964（R）（不适于 RS-485）和打印机驱动程序。

CP341 用于执行强大的点到点高速串行通信。具有不同物理特性的 3 个型号，分别是 RS-232C（V.24）、20 mA（TTY）和 RS-422/RS-485（X.27）。CP341 执行的协议有 ASCII、3964（R）、RK 512 和客户协议（可装载）。

CP343-2 是用于 SIMATIC S7-300 PLC 和 ET 200M 分布式 I/O 的 AS-Interface 主站。CP343-2 最多可连接 62 个 AS-Interface 从站，并且可集成模拟值传送（遵循扩展 AS-Interface 规范 V2.1）。支持所有 AS-Interface 主站，符合扩展 AS-Interface 接口技术规范 V2.1

CP342-5 是用于 PROFIBUS 总线系统的 SIMATIC S7-300 PLC 和 SIMATIC C7 的通信模块。CP342-5 减轻了 CPU 的通信任务。它使用的通信协议有：PROFIBUS-DP、PG/OP 通信、S7 通信等。使用 CP342-5 通信处理模块传输速度可达 9.6Kbit/s～12Mbit/s。

CP343-1 在工业以太网上独立处理数据通信，主要用于操作员之间的连接。该模块有自身的处理器，符合 OSI 模型，支持的协议有 ISO、TCP/IP、S7 通信和 PG/OP 通信等。以太网模块目前有 CP343-1 Lean、CP343-1、CP343-1 IT 和 CP343-1 PN 四大类，各有特色。

6．S7-300 PLC的功能模块

（1）计数器模块

模块的计数器均为 0～32 位或±31 位加减计数器，可以判断脉冲的方向，模块给编码器供电。达到比较值时发出中断。可以 2 倍频和 4 倍频计数。有集成的 DI/DO。

FM 350-1 是单通道计数器模块，可以检测最高达 500kHz 的脉冲，有连续计数、单向计数、循环计数 3 种工作模式。FM 350-2 和 CM 35 都是 8 通道智能型计数器模块。

其功能是除了在高速计数外，还可以在同步测量模式下工作。模块在频率测量的基础上增加了转速测量和周期测量的功能。此外在诊断中断的内容方面，增加了外部电压故障、DC5V 故障、RAM 出错、A/B/N 输入错误等检测功能。

（2）位置控制与位置检测模块

FM351 双通道定位模块用于控制变级调速电动机或变频器。FM353 是步进电机定位模块。FM354 是伺服电机定位模块。FM357 可以用于最多 4 个插补轴的协同定位。FM352 高速电子凸轮

控制器，它有 32 个凸轮轨迹，13 个集成的 DO，采用增量式编码器或绝对式编码器。

SM338 超声波传感器检测位置，无磨损、保护等级高、精度稳定不变。

（3）闭环控制模块

FM355 闭环控制模块有 4 个闭环控制通道，有自优化温度控制算法和 PID 算法。

功能模块的使用比较复杂，接线也相对复杂，请读者参考相关手册，在此不再赘述。

7．S7-300 PLC的其他模块

（1）称重模块

SIWAREX U 称重模块是紧凑型电子秤，测定料仓和贮斗的料位，对吊车载荷进行监控，对传送带载荷进行测量或对工业提升机、轧机超载进行安全防护等。

SIWAREX M 称重模块是有校验能力的电子称重和配料单元，可以组成多料称重系统，安装在易爆区域。

（2）电源模块

PS 307 电源模块将 120/230 交流电压转换为 24V 直流电压，为 S7-300 PLC、传感器和执行器供电。输出电流有 2A、5A 和 10A 共 3 种。电源模块安装在 DIN 导轨上的插槽 1。S7-300 PLC 的浮动参考电位如图 2−25 所示。

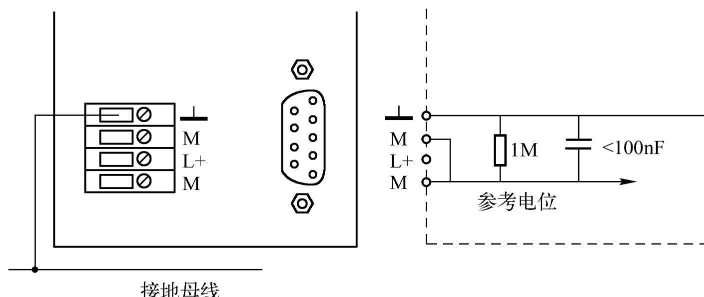


图 2−25　S7-300 PLC 的浮动参考电位

某些大型工厂（例如化工厂和发电厂）为了监视对地的短路电流，可采用浮动参考电位，可以将 M 点与接地点之间的短接片去掉。

（3）接口模块

接口模块（IM），接口模块用于多机架配置时连接主机架（CR）和扩展机架（ER）。S7-300 PLC 通过分布式的主机架和连接的扩展机架（最多可连接 3 个扩展机架），可以操作最多 32 个模块。

2.2　S7- 400 PLC常用模块及其接线

2.2.1　S7-400 PLC的概述

SIMATIC S7-400 PLC 是用于中、高档性能范围的可编程序控制器，其尺寸比 S7-300 PLC 大。采用模块化及无风扇的设计，坚固耐用，容易扩展和广泛的通信能力，容易实现分布式结构以及用户友好的操作，使 SIMATIC S7-400 PLC 成为中、高档性能控制领域中首选的理想解决方案。与 S7-300 PLC 相比，S7-400 PLC 的体积更加大，性能更加卓越，其特点如下：



1）运行速度快，S7-416 PLC 执行一条二进制指令只要 0.08μs。

2）存储器容量大，例如 CPU 417-4 的 RAM 可以扩展到 16MB，装载存储器（EEPROM 或 RAM）可以扩展到 64MB。

3）I/O 扩展功能强，可以扩展 21 个机架，CPU 417-4 最多可以扩展 262144 个数字量 I/O 点和 16384 个模拟量 I/O。

4）有极强的通信能力，集成的 MPI 能建立最多 32 个站的简单网络。大多数 CPU 集成有 PROFIBUS-DP 主站接口，用来建立高速的分布式系统，通信速率最高为 12Mbit/s。

5）集成的 HMI 服务，只需要为 HMI 服务定义源和目的地址，自动传送信息。

S7-400 PLC 的外形如图 2−26 所示。

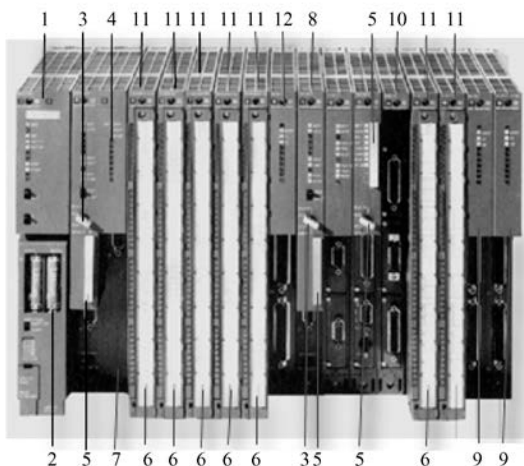


图 2−26　S7-400 PLC 的外形图

1—电源模板　2—后备电池　3—钥匙开关　4—状态和故障 LED　5—存储器卡，

6—有标签区的前连接器　7—CPU 1　8—CPU 2　9—CP 接口　10—IM 接口模板　11—I/O 模板

2.2.2　S7-400 PLC的机架

1．S7-400 PLC的机架简介

S7-400 PLC 的机架是安装所有模块的基本框架，这些模块通过背板总线进行交换数据和供电。S7-400 PLC 的机架种类和应用见表 2−3。

表 2−3　S7-400 PLC 的机架种类和应用

机架	插槽总数	可用总线	可用领域	说明
UR1	18	I/O 总线 通信总线	CR 或者 ER	适用于所有的模块类型
UR2	9			
ER1	18	受限 I/O 总线	ER	适用于 SM、IM 和 PS 模块 I/O 总线受以下控制： 1）不会响应模块中断 2）不能使用 24V 供电模块 3）模块不能使用模块的后备电源供电，也不能通过外加电源给 CPU 或接收 IM 的电压加电
ER2	9			

（续）

机架	插槽总数	可用总线	可用领域	说明
CR2	18	分段 I/O 总线 连续通信总线	分段 CR	适用于除 IM 外所有的模块，I/O 总线分两段，占 10 槽和 8 槽
CR3	4	I/O 总线 通信总线	标准系统 CR	适用于除 IM 外所有的模块，CPU41X-H 仅限单机操作
UR2-H	2×9	分段 I/O 总线 连续通信总线	为紧凑安装容错型系统细分为 CR 或者 ER	适用于除 IM 外所有的模块，I/O 总线分两段，各占 9 槽

1）机架的数据交换。I/O 总线（P 总线）是机架的并行背部总线，用于 I/O 信号交流，对于信号模块的过程数据也是通过 I/O 总线进行。通信总线（C 总线）是机架背部串行总线，用于快速交换 I/O 信号相关的大量数据。除 ER1 和 ER2 外，其他的机架只有一条通信总线。

2）机架的供电。通过背部总线和基本连接器，由安装在机架最左侧的电源模块为机架上的模块提供所需的工作电压（5V 用于逻辑控制，24V 用于接口模块）。对于本地连接，还可以通过 IM460-1/IM461-1 接口模块为 ER 供电。IM460-1 有两个接口，每个接口最多可以通过 5A 的电流，可以为每个 ER 提供 5A 的电流。

2．UR1 机架（通用机架）

UR1 和 UR2 机架用于装配中央控制器和扩展单元。UR1 和 UR2 机架都有 I/O 总线和通信总线。UR1 最多可容纳 18 个模块，UR2 最多可容纳 9 个模块。UR1 机架如图 2-27 所示。

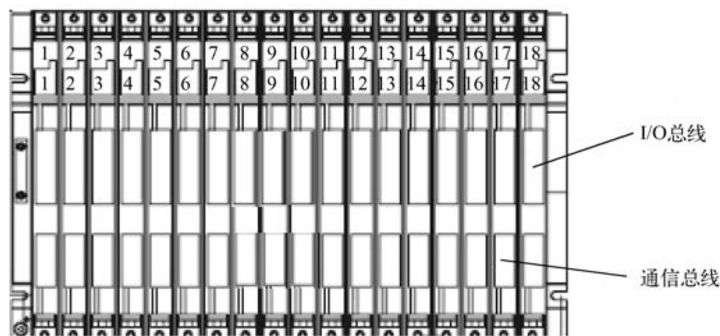


图 2-27　UR1 机架及其总线

3．CR2 和CR3 机架

CR2 机架用于分段式中央机架设计，可安装中央控制器和扩展单元。CR2 有 I/O 总线和通信总线。I/O 总线分为两个局部总线区段，分别为 10 个和 8 个插槽，如图 2-28 所示，I/O 总线分为两个局部总线区段①和②，其中区段①为 10 槽，区段②为 8 槽，③为通信总线。CR2 共 18 槽，可安装一个电源模块和两个 CPU 模块和其他模块。

CR3 是 4 槽的中央机架，有 I/O 总线和通信总线各一条。



4．UR2-H机架

UR2-H 机架用于在一个机架上装配两个中央控制器或扩展单元。 UR2-H 机架实质上代表同一装配导轨上的两个电隔离的 UR2 机架。UR2-H 主要应用于紧凑结构的冗余 S7-400 H PLC 系统（同一机架上有两个设备或系统）。

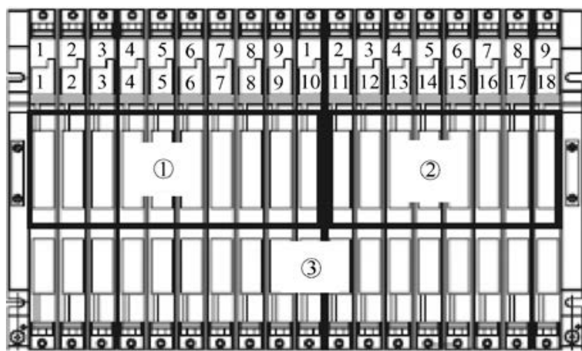


图 2−28　UR2 机架及其总线

UR2-H 机架及其总线如图 2−29 所示，机架分为①和②两段，每段 9 槽，每段都有各自的电源和 CPU。

① 段的 1～9 号槽按顺序分别安装电源模块、CPU 模块、DI 模块等。

② 段的 1～9 号槽按顺序分别安装电源模块、CPU 模块、DI 模块等。

注意西门子机架的槽位从 1 开始，而有的 PLC 机架的槽位从 0 开始（如 GE 的 RX3i）。

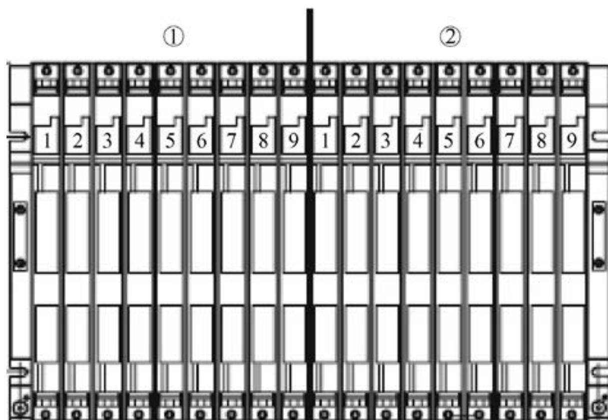


图 2−29　UR2-H 机架及其总线

2.2.3　S7-400 PLC常用模块及其连接

S7-400 PLC 系统的配置比较灵活，不同的配置可以选用不同的模块。S7-400 PLC 常用

的模块有电源模块、CPU、接口模块 IM、数字量输入模块 DI、数字量输出模块 DO、模拟量输入模块 AI、模拟量输出模块 AO、功能模块 FM 和通行处理模块 CP 等。

1．CPU模块

CPU 模块是控制系统的核心，负责系统的中央控制，存储并执行程序，实现通信功能。S7-400 PLC CPU 共有 7 款 CPU，另外还有 S7-400H/F/FH PLC 的 CPU。

1）CPU 412-1 和 CPU 412-2 用于中等性能范围的小型控制系统。CPU 412-1 有 96KB 的 RAM，最大可扩展 32KB 的 DI/DO 和 2KB 的 AI/AO，有一个组合式的 MPI/DP 接口，允许 PROFIBUS-DP 操作。CPU 412-2 有 144KB 的 RAM，最大可扩展 32KB 的 DI/DO 和 2KB 的 AI/AO，有一个组合式的 MPI/DP 接口，允许 PROFIBUS-DP 操作。此外，这两款 CPU 的运行速度和扩展能力也有所不同。

2）CPU 414-2 和 CPU 414-3 适合于中等性能系统的控制。它们满足对程序规模和指令处理速度及复杂通信的更高要求。CPU 414-2 有 256KB 的 RAM，最大可扩展 64KB 的 DI/DO 和 4KB 的 AI/AO，有一个组合式的 MPI/DP 接口和一个 PROFIBUS-DP 接口，允许 PROFIBUS-DP 操作。CPU 414-3 有 768KB 的 RAM，最大可扩展 64KB 的 DI/DO 和 4KB 的 AI/AO，有一个组合式的 MPI/DP 接口、一个 PROFIBUS-DP 接口和一个 IF 模块插槽，允许 PROFIBUS-DP 操作。这两款 CPU 功能较强，具有集成 PROFIBUS-DP 接口，能作为主站直接连接到 PROFIBUS-DP 总线。

3）CPU 416-2 和 CPU 416-3 应用于高性能范围中的各种高要求场合。CPU 416-2 有 1.6MB 的 RAM，最大可扩展 128KB 的 DI/DO 和 8KB 的 AI/AO，带有一个组合式的 MPI/DP 接口和一个 PROFIBUS-DP 接口。CPU 416-3 有 3.2MB 的 RAM，最大可扩展 128KB 的 DI/DO 和 8KB 的 AI/AO，有一个组合式的 MPI/DP 接口、一个 PROFIBUS-DP 接口和一个 IF 模块插槽。这两款 CPU 功能强大，具有集成 PROFIBUS-DP 接口，能作为主站直接连接到 PROFIBUS-DP 总线。

4）CPU 417-4 适用于更高性能范围的最高要求场合。其在 S7-400 PLC CPU 中是功能最强大的。具有集成 PROFIBUS-DP 接口，能作为主站直接连接到 PROFIBUS-DP 总线。有两个槽适用于 IF 模块（串行）。

5）CPU 417-4H 是 S7-400H PLC 和 S7-400F/FH PLC 中功能最强的 CPU，是硬件冗余 CPU。可配置为容错式 S7-400H PLC 系统。连接上 F 运行授权后，可以作为 S7-400F/FH PLC 容错自动化系统应用。集成的 PROFIBUS-DP 接口能作为主站直接连接到 PROFIBUS-DP 现场总线。

S7-400H PLC 采用“热备用”模式的主动冗余原理，在发生故障时，无扰动地自动切换。两个控制器使用相同的用户程序，接收相同的数据，两个控制器同步地更新内容，任意一个子系统有故障时，另一个承担全部控制任务。

6）S7-400F PLC。安全型自动化系统，出现故障时转为安全状态，并执行中断。S7-400FH PLC 是安全及容错自动化系统，如果系统出现故障，生产过程能继续执行。

2．数字量输入模块SM421 及其连接

数字量输入模块将二进制过程信号连接到 S7-400 PLC，也就是将按钮、接近开关等数字量连接到 S7-400 PLC。从输入电信号的种类分，数字量输入模块主要有 3 种：即直



流输入、交流输入和交直流输入。目前数字量输入模块 SM421 有 7 个规格。以下将介绍有代表性的几种。

数字量输入模块（6ES7 421-1BL01-0AA0）内部结构和连接如图 2−30 所示，很明显 SM421 模块的内部具有光电隔离，模块是 24V 直流 PNP 输入（高电平有效），其连接与 SM321 的 PNP 输入模块的连接类似。

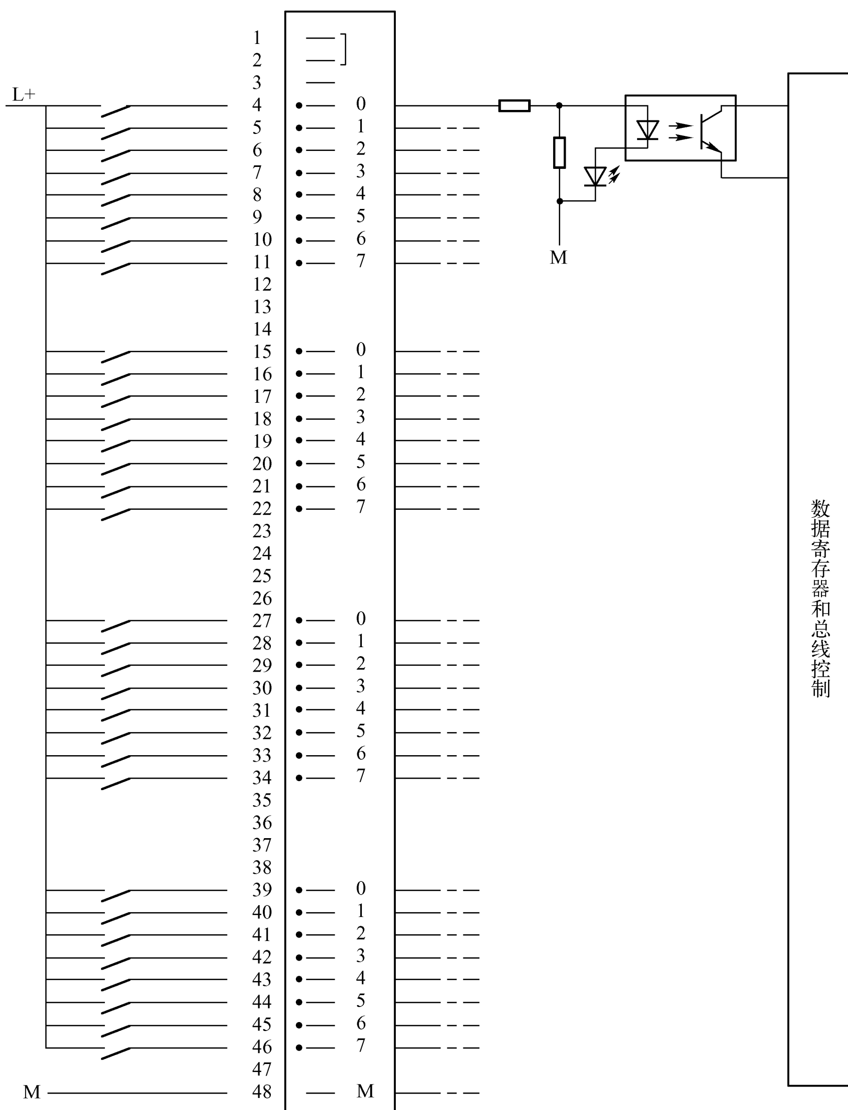


图 2−30　数字量输入模块（6ES7 421-1BL01-0AA0）内部结构和连接

数字量输入模块（6ES7 421-5EH00-0AA0）内部结构和连接如图 2−31 所示， SM421 模块是交流输入，额定电压为 AC120 V，可连接普通开关（按钮）和两线接近开关。

数字量输入模块（6ES7 421-1FH00-0AA0）内部结构和连接如图 2-32 所示， SM421 模块是交直流 16 点输入，额定电压为 UC120/230 V（UCV 代表交流和直流信号输入均可），可连接普通开关（如按钮）和两线接近开关。

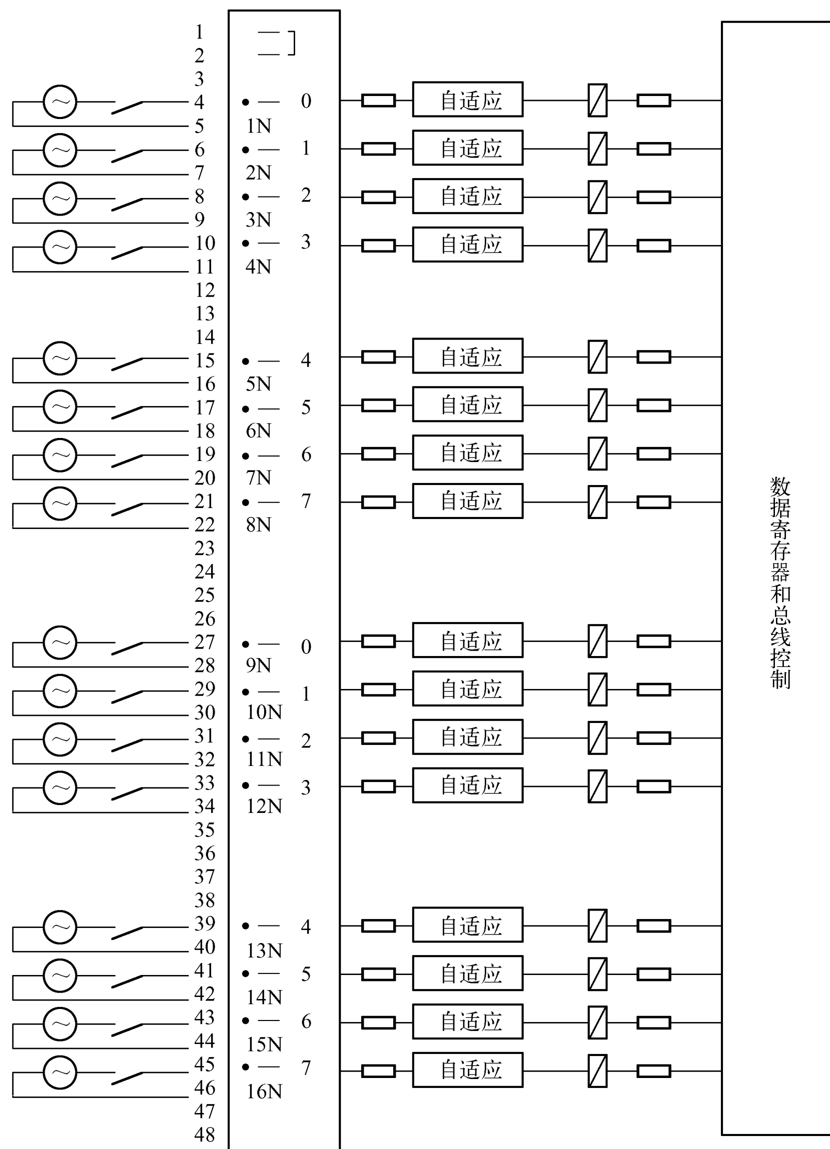


图 2-31　数字量输入模块（6ES7 421-5EH00-0AA0）内部结构和连接

3．数字量输出模块SM422 及其连接

数字量输出模块将二进制过程信号连接到 S7-400 PLC，通过这些模板，将执行器连接到 S7-400 PLC，一般常见的是电磁阀、接触器、指示灯和变频器等。从输出负载



的种类分，数字量输出模块主要有 3 种：即晶体管型（直流输出）、晶闸管型（交流输出）和继电器型（交直流输出）。目前数字量输出模块 SM422 有 8 个规格，以下将介绍有代表性的几种。

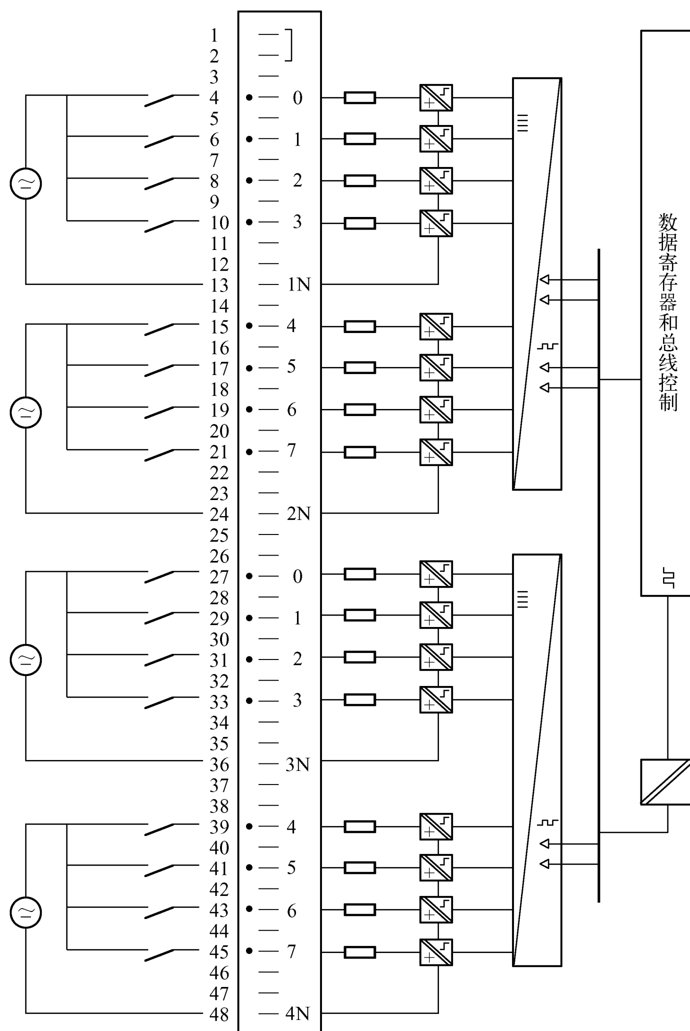


图 2−32　数字量输入模块（6ES7 421-1FH00-0AA0）内部结构和连接

数字量输出模块（6ES7 422-1BH11-0AA0）内部结构和连接如图 2−33 所示，这个 SM422 模块晶体管型，共 16 点，分两组，每组 8 点，额定电压为 DC24 V。调试模块时，不需要给各组的 8 个输出提供负载电压（例如，1L+和 3L+），即使仅用 L+为单个组供电，模块也能完全正常工作。但正常工作时，还是要将 1L+、2L+和 3L+并联在 DC24 V 上，1M、2M 和 3M 并联在 0 V 上。

数字量输出模块（6ES7 422-1FF00-0AA0）内部结构和连接如图 2−34 所示，这个 SM422 模块晶闸管型，共 8 点，分 8 组，每组 1 点，额定电压为 AC120/230 V。

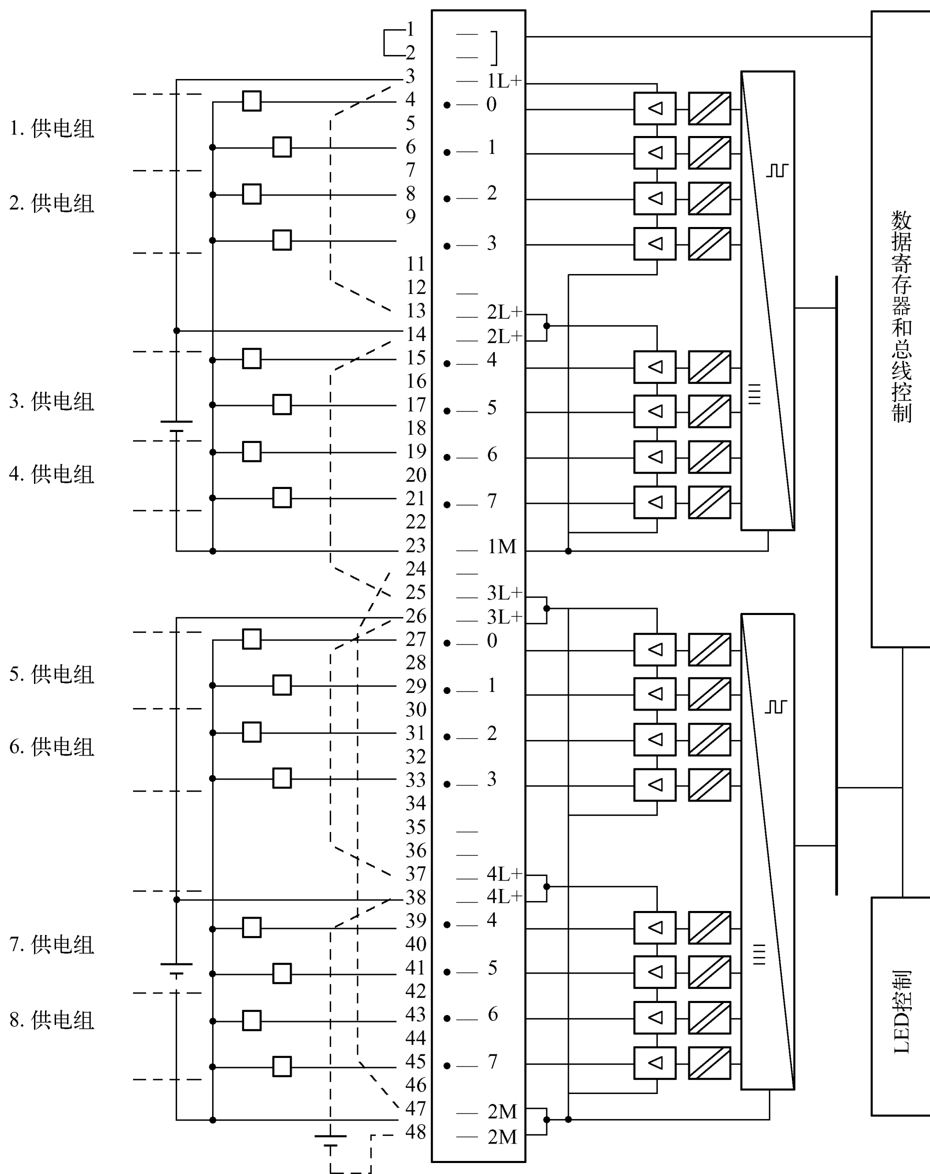


图 2−33　数字量输出模块（6ES7 422-1BH11-0AA0）内部结构和连接

数字量输出模块（6ES7 422-1HH00-0AA0）内部结构和连接如图 2−35 所示，SM422 模块是继电器型，共 16 点，分 8 组，每组 2 点，输出电流为 5 A，额定负载电压为 AC230 V/DC125 V。由于是继电器输出，所以电源的接入无极性之分，不必像晶体管输出那



样有极性之分。

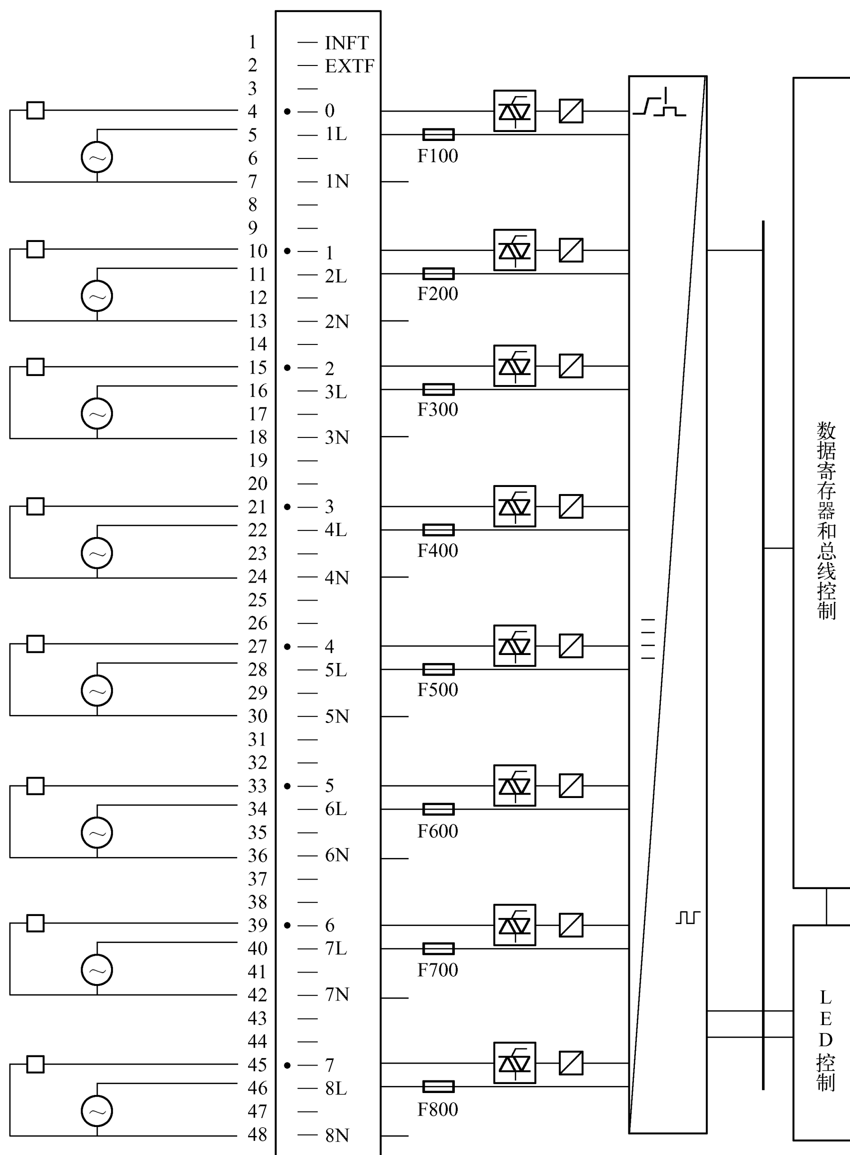


图 2-34　数字量输出模块（6ES7 422-1FF00-0AA0）内部结构和连接

4．模拟量输入模块SM431 及其连接

模拟量输入模块 SM431 用于 S7-400 PLC 模拟量输入，用于连接电压、电流、热电偶、电阻器和热电阻，分辨率从 13 到 16 位。目前模拟量输入模块 SM431 有 7 个规格。以下将介绍具有代表性的一种。

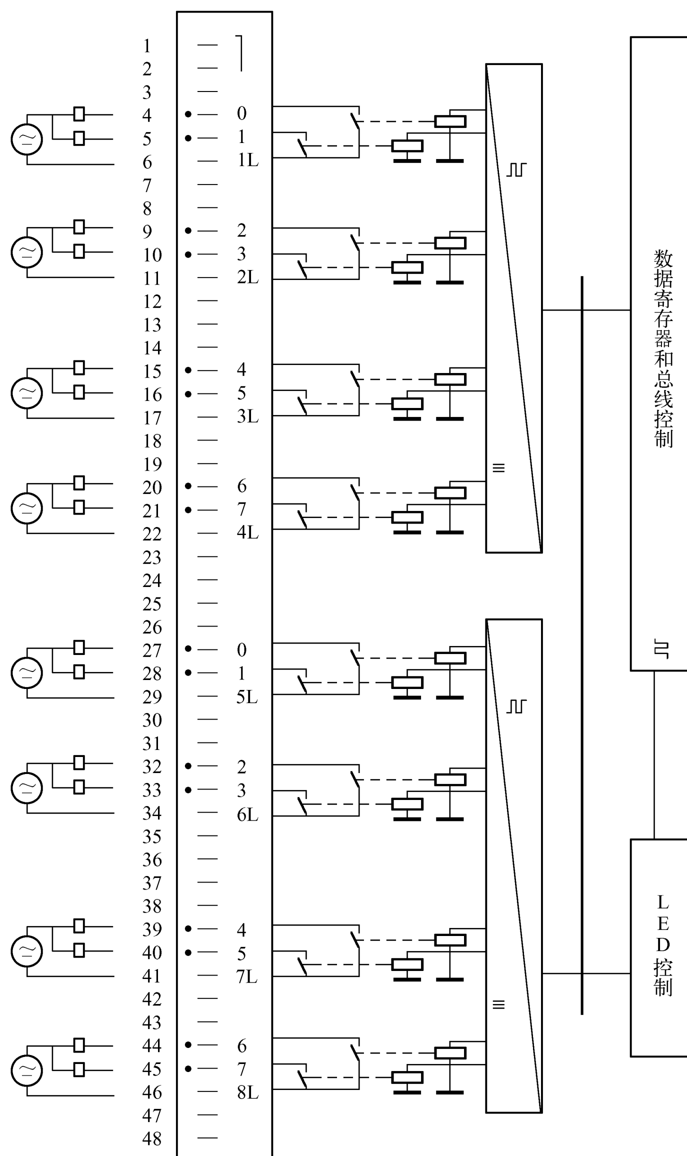


图 2-35　数字量输出模块（6ES7 422-1HH00-0AA0）内部结构和连接

模拟量输入模块 SM431（6ES7 431-1KF00-0AB0）简称 AI 8×13 位，它有 8 个输入点用于电压/电流测量，4 个输入点用于电阻测量，可并行调整的各种测量范围，精度 13 位，模拟部分与 CPU 隔离，通道之间以及连接传感器的参考电位与 MANA 之间允许的最大共模电压为 AC30 V。SM431 的连接如图 2-36 所示。若第一二通道都用于测量电压信号，那么端子 6 和 9 以及端子 11 和 14 分别与 2 路电压信号连接；若第一二通道都用于测量电流信号，那么端子 6 和 7 短接，端子 11 和 12 短接，端子 8 与 9 以及端子 13 与 14 分别与 2 路电流信号连接；若第一二通道都用于测量电阻，那么只能测量一路信号，先将热电阻的一端的两个引线分别连接端子 6 和 11 上，再将热电阻的另一端的引线连到端子 9 和 14 上。



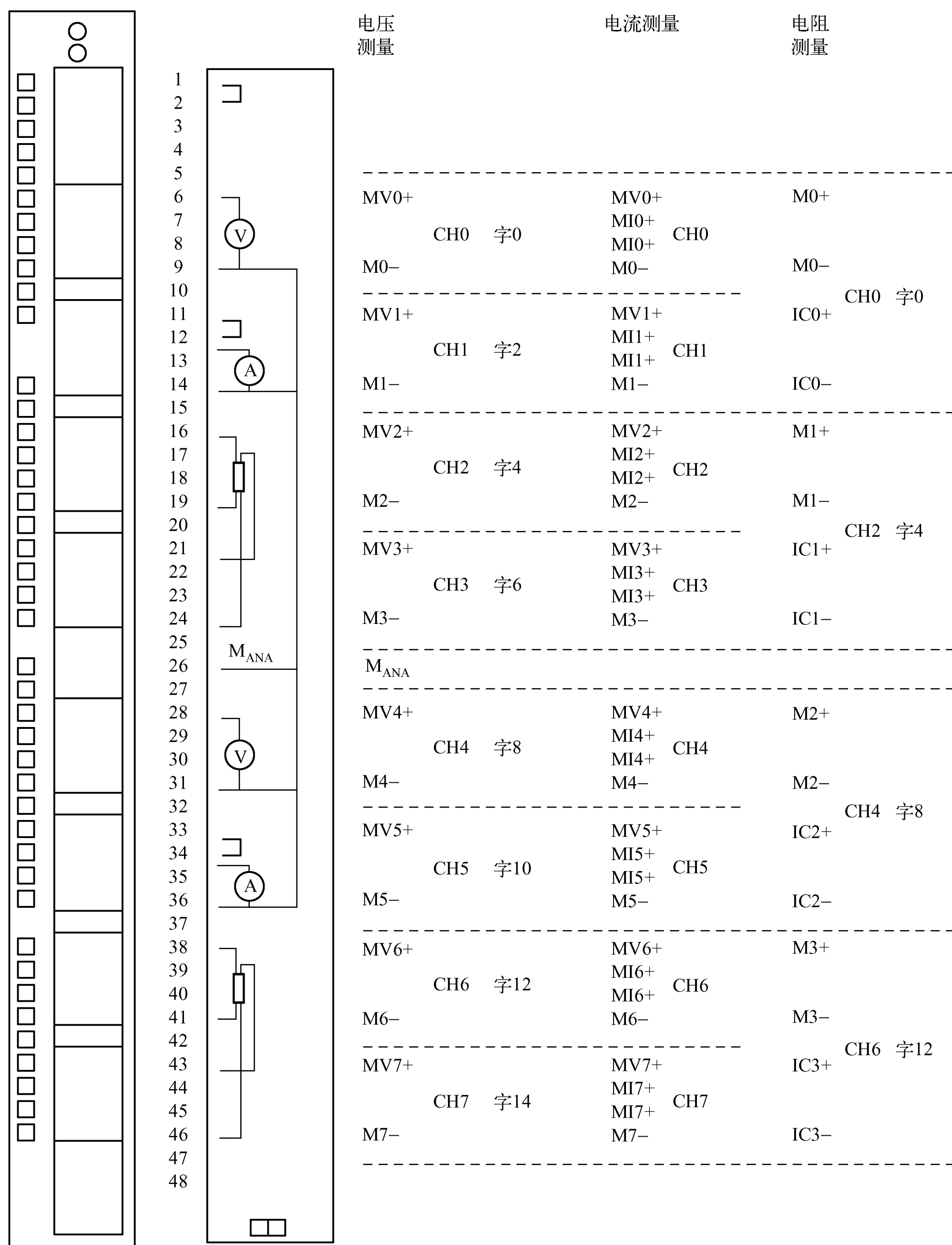


图 2-36　模拟量输入模块（6ES7 431-1KF00-0AB0）的连接

5．模拟量输出模块SM432 及其连接

模拟量输出模块 SM432 用于 S7-400 PLC 模拟量输出，连接模拟量执行器。目前模拟量输出模块 SM432 有 1 个规格。

模拟量输出模块 SM432（6ES7 432-1HF00-0AB0），简称 AO8×13 位，它有 8 点输出点，其输出通道可编程为电压输出或者电流输出（有关设置方法在后面的章节中会讲解），

输出精度 13 位，模拟量部分与 CPU 和负载电压隔离，通道之间或通道与 MANA 之间允许的最大共模电压为 30 V。SM432 的连接如图 2-37 所示。

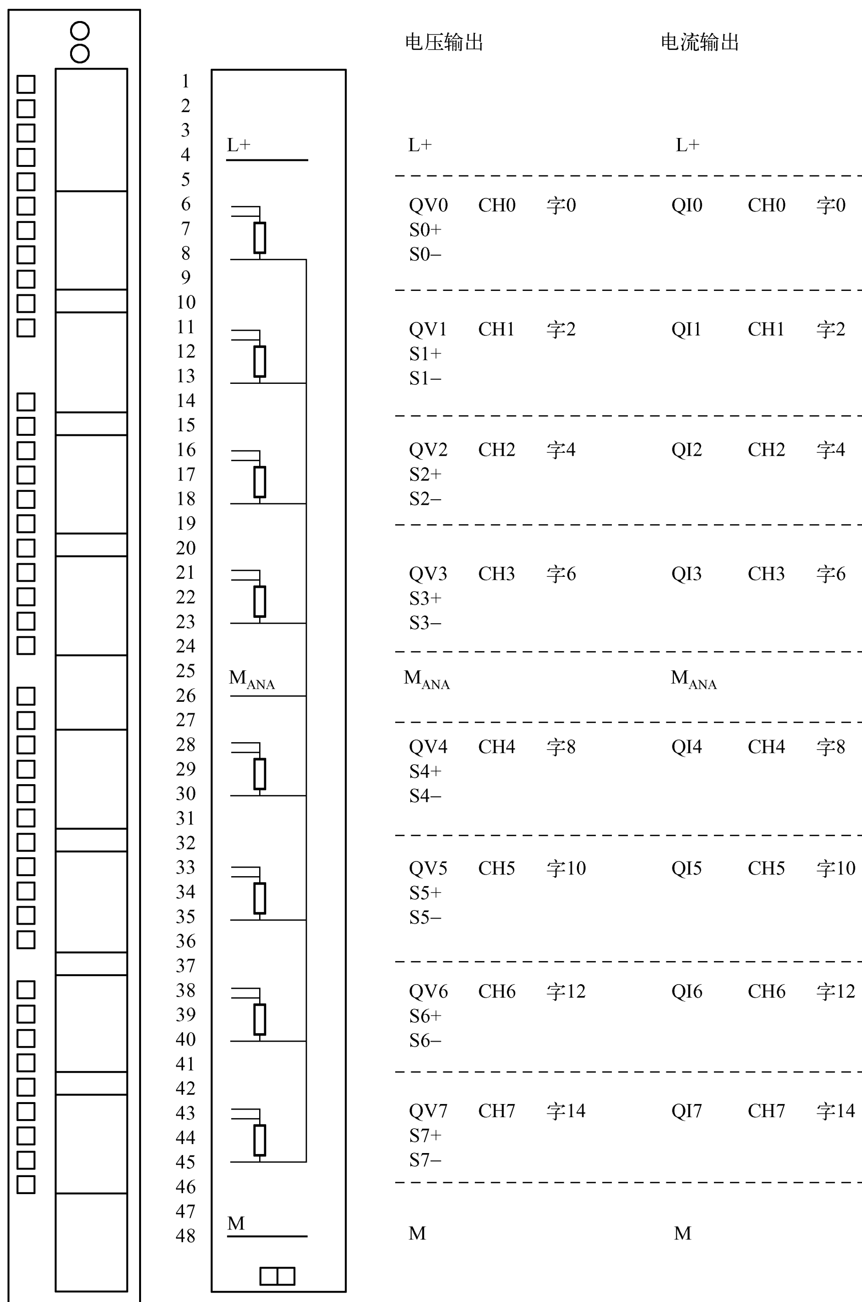


图 2-37　模拟量输出模块（6ES7 432-1HF00-0AB0）的连接

6．电源模块

S7-400 PLC 的电源有 PS405 和 PS407 两大类。电源模块用于对 S7-400 PLC 的供电，具体为：



1）用于将 AC 或 DC 网络电压转换为所需的 DC5 V 和 DC24 V 工作电压 输出电流为 4A、10 A 和 20 A。

2）通过 IM 向扩展机架供电。电源的位置安装在机架的最左边，也就说从第一槽开始安装，根据配置不同可占用 1 至 3 槽。

PS405 和 PS407 是有区别的，主要是接入输入电压的种类不同，PS405 电源模块输入电压为 DC19.2V 到 DC72 V 的直流电压，并在次级侧提供 DC5V 和 DC24V 电压。而 PS407 电源模块输入电压为 AC85V 到 AC264 V 线路电压或 DC88V 到 DC300 V 电压，并在二级侧提供 DC5V 和 DC24 V 电压。

7．通信处理模块

S7-400 PLC 常用的通信处理模块有如下几种：

① CP 440 通信处理器。用于利用 RS 422/RS 485（X.27）进行短报文帧的高性能传输场合。

② CP 441-1 和 CP 441-2。通过点对点链接进行高速大容量串行数据交换。

③ CP 443-5。用于 PROFIBUS 通信。

④ CP 443-1 和 CP 443-1 IT。用于将 SIMATIC S7-400 PLC 连接到以太网上。

⑤ CP 444。应用 MMS 服务，根据 MAP3.0，连接工业以太网。用于减轻 CPU 的通信任务和实现深层的连接。

8．功能模块

S7-400 PLC 常用的功能模块有计数模块 FM450-1、定位模块 FM451 和 FM453、电子凸轮控制器 FM452 和闭环模块 FM455 等。

9．接口模块

S7-400 PLC 的接口模块分为两种：发送和接收模块。常用的发送模块有 IM460-0、IM460-1、IM460-2、IM460-3 和 IM460-4。常用的接收模块有 IM461-0、IM461-1、IM461-3 和 IM461-4。

小结

重点难点总结

1．西门子 PLC 的 5 大系列产品应用范围。

2．常用 S7-300/400 PLC 的 CPU 模块、数字量输入/输出模块、模拟量输入/输出模块、功能模块、机架和电源模块的功能。

3．常用 S7-300/400 PLC 的 CPU 模块、数字量输入/输出模块、模拟量输入/输出模块和电源模块的接线和使用注意事项。

4．常用 S7-300/400 PLC 模块的安装。

习题

1．西门子 PLC 产品分为哪几大系列，分别使用在什么场合？

2．S7-300 PLC 主要由哪几类模块组成？各类模块有什么功能？

3．S7-400 PLC 主要由哪几类模块组成？各类模块有什么功能？

4．使用 S7-300 PLC 的模拟量输入模块要注意哪些问题？

5．S7-300 PLC 的数字量输入模块主要分哪几类？

6．如果一个系统选用的接近开关是 NPN 型，选用什么类型的数字量输入模块比较合适？若选用是 PNP 型接近开关呢？

7．S7-300 PLC 的数字量输出模块主要分哪几类？

8．使用 S7-300 PLC 的模拟量输出模块要注意哪些问题？

9．若将模拟量输入模块（6ES7 331-7PF01-0AB0）的 IC0+与 IC0-短接会产生什么问题，为什么？

10．若一个系统既有交流负载，又有直流负载，选用什么类型的数字量输出模块比较合适？

11．S7-300 PLC 与 S7-400 PLC 的编址方式有什么不同？

12．一个控制系统有 14 点数字量输入，10 点数字量输出，3 点模拟量输入，1 点模拟量输出，若选用 S7-300 PLC，请选用合适的输入/输出模块，并分配 I/O 地址。

第3章

STEP 7 软件使用入门

本章介绍 S7-300/400 PLC 可编程序控制器的编译软件 STEP 7 的使用方法，并介绍使用 STEP 7 编译一个简单程序的完整过程的例子，这是学习本书后续内容的必要准备。

3.1 STEP 7 简介

3.1.1 初识STEP 7

STEP 7 是一种用于对 SIMATIC 可编程序逻辑控制器进行组态和编程的标准软件包。它是 SIMATIC 工业软件的一部分。STEP 7 标准软件包有下列各种版本：

1）STEP 7 Micro/DOS 和 STEP 7 Micro/Win。用于 SIMATIC S7-200 上的简化版单机应用程序。

2）STEP 7。用于 SIMATIC S7-300 PLC/S7-400 PLC、SIMATIC M7-300/M7-400 以及 SIMATIC C7，标准 STEP 7 软件包提供一系列应用程序，具体如下。

① SIMATIC 管理器（SIMATIC Manager）可以集成管理一个自动化项目的所有数据，可以分布式地读/写各个项目的用户数据。其他的工具都可以在 SIMATIC 管理器中启动。

② 符号编辑器。其可以管理所有的共享符号。

③ 诊断硬件。其功能可以提供可编程序控制器的状态概况。其中可以显示符号，指示每个模块是否正常。

④ 编程语言。其用于 S7-300/400 PLC 的编程语言梯形图（Ladder Logic）、语句表（Statement List）和功能块图（Function Block Diagram）都集成在一个标准的软件包中。此外还有 4 种语言作为可选软件包使用，分别是 S7 SCL（结构化控制）编程语言、S7 Graph（顺序控制）编程语言、S7 Hi Graph（状态图）编程语言和 S7 CFC（连续功能图）编程语言。

⑤ 硬件组态。其工具可以为自动化项目的硬件进行组态和参数配置。可以对机架上的硬件进行配置，设置其参数及属性。

⑥ 网络组态。其工具用于组态通信网络连线，包括网络连接参数设置和网络中各个通信设备的参数设定，选择系统集成的通信或功能块，可以轻松实现数据的传送。

本书使用的软件是 STEP 7 V5.4 SP4+HF3Chinese。STEP 7 V5.4 SP4+HF3Chinese 的大部分界面已经汉化，非常适合对外语不熟悉的人员使用。STEP 7 具有使用简单、面向对象、直观的用户界面、组态取代编程、统一的数据库、超强的功能、编程语言符合 IEC 1131-3、基于 Windows 操作系统。

使用 STEP 7 的基本软硬件条件是：一台西门子编程设备或者一台个人计算机、STEP 7 软件包和相应的许可证密匙，一台 S7-300/400 PLC 或者 S7-PLCSIM。

3.1.2　安装STEP 7 注意事项

1）安装 STEP 7 的操作系统可以是：Microsoft Windows 2000 或 Windows XP、Windows Server 2003（从 STEP 7 V5.4 SP3 开始，也支持 Windows Vista 32 Business 和 Ultimate 操作系统）。不支持 HOME 版操作系统。

2）安装 STEP 7 的基本硬件要求，包含下列各项的编程设备或 PC：

- 奔腾处理器（600 MHz）。
- 至少 512MB RAM。
- 彩色监视器、键盘和鼠标，Microsoft Windows 支持所有这些组件。

3）最好关闭监控和杀毒软件。

4）软件的存放目录中不能有汉字。例如将软件存放在“C:/软件/STEP 7”目录中就不能安装。

3.2　编程界面的SIMATIC管理器

在 STEP 7 中，用项目来管理一个自动化系统的硬件和软件。STEP 7 用 SIMATIC 管理器对项目进行集中管理，它可以方便地浏览 SIMATIC S7、C7 和 WinAC 的数据。因此，掌握项目创建的方法就非常重要。

3.2.1　创建项目

1．使用向导创建项目

首先双击桌面上的“STEP 7”图标，进入 SIMATIC Manager 窗口，进入主菜单“文件”，选择“新建项目”向导，弹出标题为“STEP 7 向导：新建项目”的对话框，如图 3-1 所示。



图 3-1　新建项目（1）

单击“下一步”按钮，在新项目中选择 CPU 模块的型号为“CPU 314C-2DP”，如图 3-2 所示。



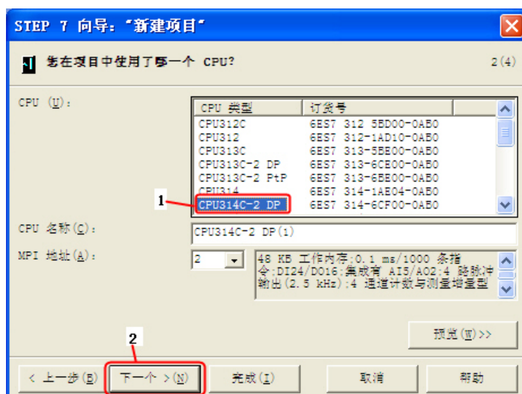


图 3-2　新建项目（2）

单击“下一步”按钮，选择需要生成的逻辑块，至少需要生成作为主程序的组织块 OB1 和编程语言 LAD（梯形图），如图 3-3 所示。

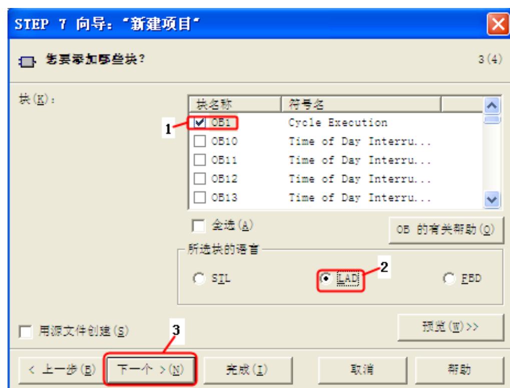


图 3-3　新建项目（3）

单击“下一步”按钮，输入项目的名称，单击“完成”按钮，生成项目，如图 3-4 所示。也可以单击工具栏上的□按钮，新建项目。

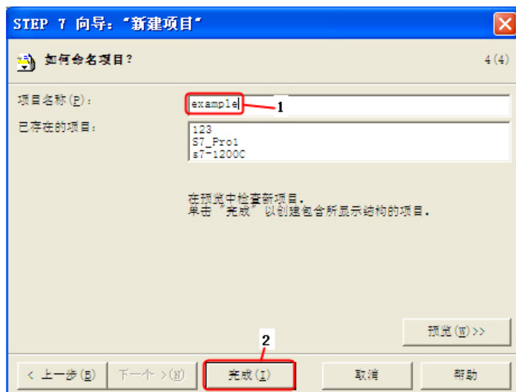


图 3-4　新建项目（4）

2．直接创建项目

进入主菜单“文件”，选择“新建”按钮，将出现如图 3-5 所示的对话框，在该对话框中分别输入“名称”、“存储位置”（路径）等内容，最后单击“确定”按钮，完成一个空项目的创建，如图 3-6 所示，这个项目中没有硬件、块等内容，需要组态硬件。

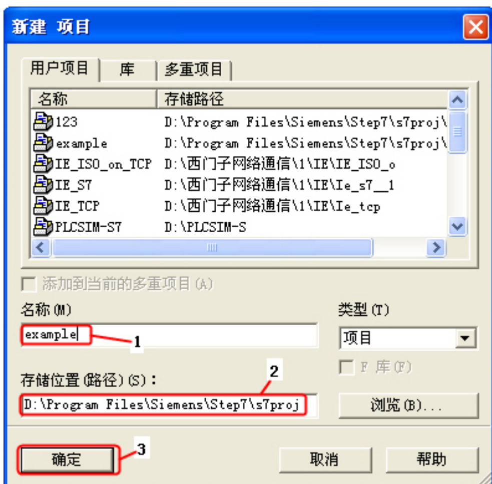


图 3-5　新建项目（5）

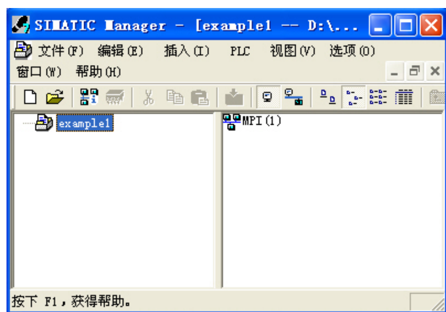


图 3-6　新建的项目（6）

3.2.2　编辑项目

1．打开已有的项目

要打开已有的项目，选择菜单栏的“文件”→“打开”，然后选择一个项目，单击“确定”按钮，打开已有的文件。例如要打开以上创建的“example”，界面如图 3-7 和图 3-8 所示。

也可以单击工具栏上的按钮，打开已有的项目。



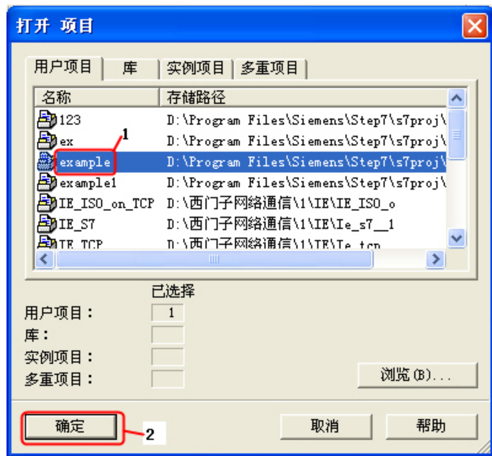


图 3-7　打开项目（1）　　图 3-8　打开项目（2）



2．复制项目

复制项目的步骤是：选中要复制的项目，在 SIMATIC 管理器中选择菜单命令“文件”→“另存为”，然后起一个新名称，单击“确定”按钮。例如要复制以上创建的“example”，如图 3-9 和图 3-10 所示。



图 3-9　复制项目（1）

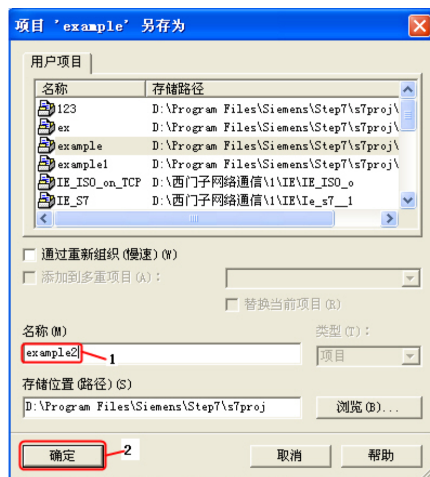


图 3-10　复制项目（2）

3．删除项目

删除项目的步骤是：在 SIMATIC 管理器中选择菜单命令“文件”→“删除”，然后选择要删除的项目，单击“确定”按钮。删除后的文件不能在回收站中找到。例如要删除以上创建的“example2”，如图 3-11 和图 3-12 所示。



图 3-11　删除项目（1）

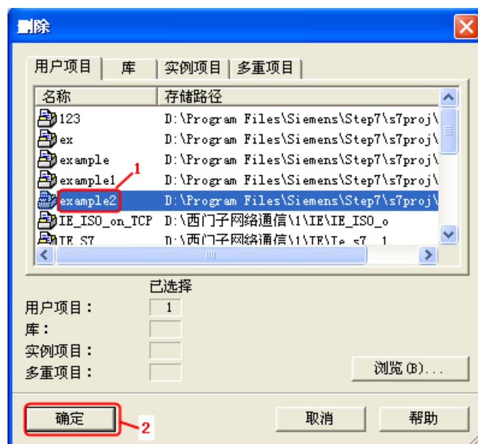


图 3-12　删除项目（2）

若要删除项目的一部分也比较容易，只要选中要删除的部分，单击鼠标右键，再单击“删除”菜单即可，例如要删除项目中的“块”，如图 3-13 所示。也可以选中要删除的部

分，按下键盘上的〈Delete〉键。

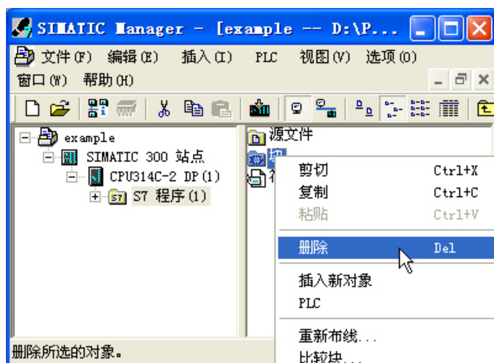


图 3-13　删除部分项目

3.3　硬件组态与参数设置

3.3.1　硬件组态

1．硬件组态的任务和步骤

在 PLC 控制系统设计前期，应根据控制系统的要求以及系统的输入/输出信号的性质和点数确定 PLC 的硬件配置，如 CPU 模块与扩展模块的型号和数量等，是否需要通信处理器模块以及种类和型号，是否需要扩展机架等。在完成上述工作后，还需要在 STEP 7 中完成硬件配置工作。

硬件组态的主要工作是根据实际系统的硬件配置，在 STEP 7 中模拟真实的 PLC 硬件配置，将电源模块、CPU 模块、信号模块、通信模块等设备安装到模拟生成的相应机架上，生成一个与真实系统完全相同的系统，并对每个硬件组成模块进行参数设置和修改的过程。S7-300/400 PLC 的模块在出厂时已经设置默认的参数作为模块的运行参数，一般情况下，用户可以不对参数进行重新设置，这样加快了硬件组态过程。当用户确实需要修改模块的参数，需要设置网络通信等工作时，都需要硬件组态。

硬件组态的基本步骤是：生成站点→生成机架并在机架上放置模块→设置模块参数→保存参数并将它下载到 PLC 中。

2．硬件组态举例

以下创建一个项目“example2”，并对此项目进行组态。

1）打开软件 STEP 7。双击 SIMATIC Manager 图标，打开 STEP 7 软件。

2）新建项目。单击工具栏上的□按钮，弹出界面如图 3-14 所示，在“名称”中输入“example2”，单击“确定”按钮。

3）插入站点。选中管理器中的项目名称“example2”，再单击菜单栏中的“插入”→“站点”→“SIMATIC 300 站点”，如图 3-15 所示。

4）启动硬件组态界面。单击项目名称“example2”左侧的“＋”，选定“SIMATIC300（1）”→“双击硬件”，之后弹出硬件组态界面，如图 3-16 所示。



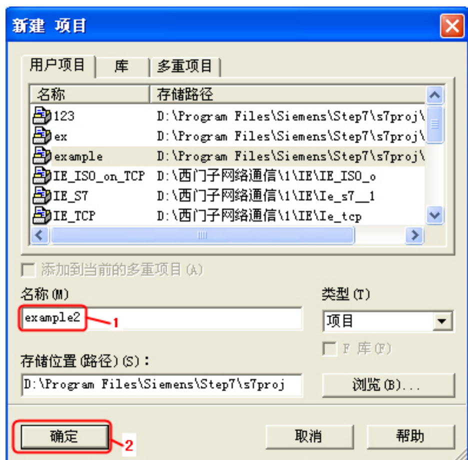


图 3-14　新建项目



图 3-15　插入站点

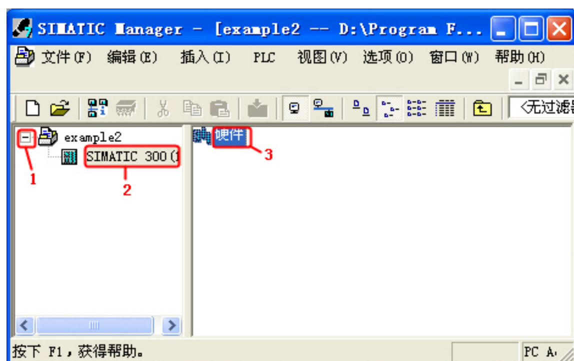


图 3-16　启动硬件组态

5）插入机架。在硬件组态界面中，单击机架“Rail”，机架自动弹出，如图 3-17 所示。

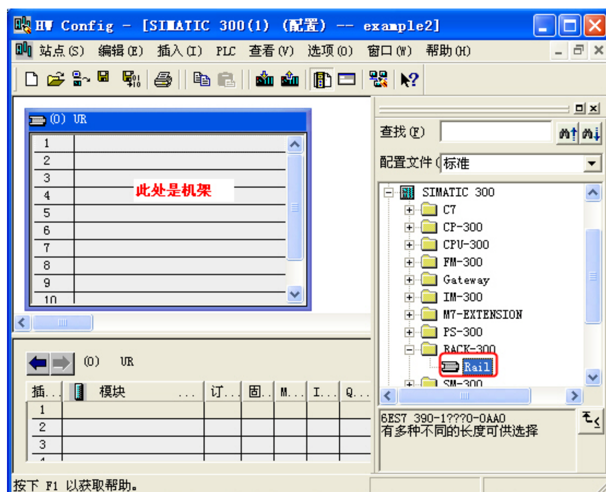


图 3-17　插入机架

6）插入电源模块。在硬件组态界面中，先选中“1”处，再双击“PS 307 5A”，电源模块自动安装到 1 号槽位，如图 3-18 所示。也可以用鼠标的左键选中“PS 307 5A”，并按住不放，将电源模块拖至 1 号槽位。

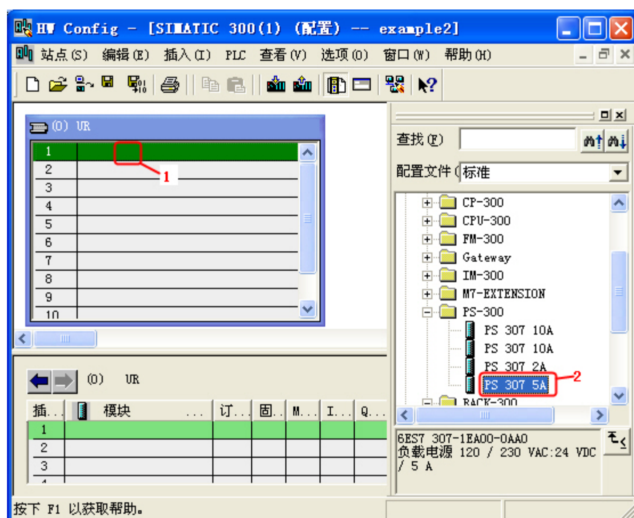


图 3-18　插入电源模块

7）插入 CPU 模块。在硬件组态界面中，用鼠标的左键选中“V2.6”，并按住不放，将 CPU 模块拖至 2 号槽位，如图 3-19 所示。也可以先选中“1”处，再双击“CPU 314-2 DP”下的“V2.6”，CPU 模块自动安装到 2 号槽位。

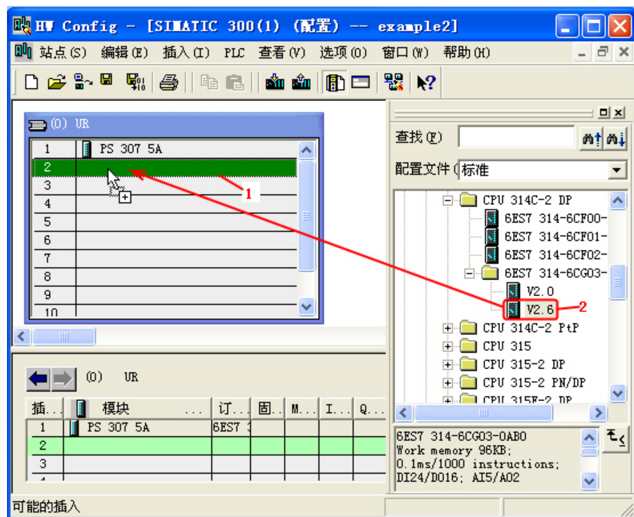


图 3-19　插入 CPU 模块（用拖放方法）

8）保存和编译组态。至此一个简单的项目已经组态完成，只要单击工具栏上的“保存和编译”按钮即可，如图 3-20 所示。



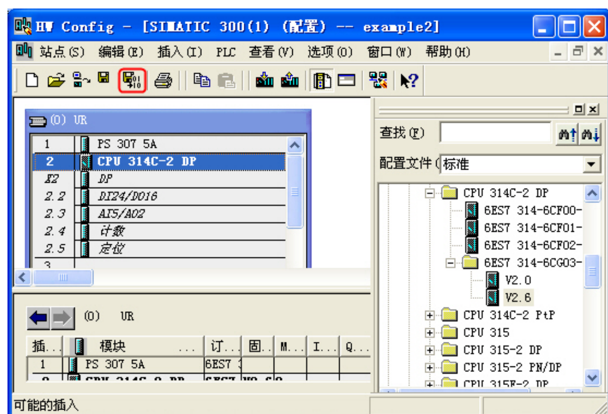


图 3-20　保存和编译组态

3.3.2　参数设置

利用 STEP 7 除了可以完成西门子 S7-300/400 PLC 的硬件组态外，还可以用来设置模块的参数。

1．CPU参数的设置

打开 HW Config 硬件组态界面，双击 CPU 模块所在的行，便可弹出属性窗口，选择某一选项卡，便可对其相应的属性进行设置。

（1）CPU 的启动参数设置

CPU 的启动性能参数可以在“属性”窗口中的“启动”选项卡中设置，如图 3-21 所示。

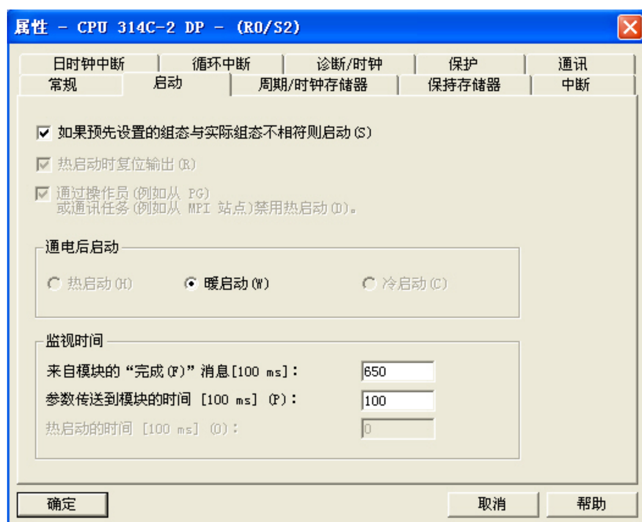


图 3-21　CPU 的属性窗口—启动参数设置

“启动”选项卡中有“如果预先设置的组态与实际组态不相符合则启动”复选框，若选中这个，则方框中应该有“√”，则当模块没有插在组态时指定的槽位或者某个槽位实际插

入的模块与组态的模块不相符合时，CPU 仍然会启动。若没有选中这个选项，则当出现上述情况时，CPU 将进入 STOP 状态。

“热启动时复位输出”和“禁止操作员热启动”选项仅用于 S7-400 PLC，在 S7-300 PLC 中该选项是灰色的。

“通电后启动”用于设置电源接通后的启动选项，可以选择单选按钮“热启动”、“暖启动”和“冷启动”。CPU-318 和 CPU 417-4 具有冷启动方式。冷启动方式启动时，所有的过程映像区和标志存储器、定时器和计数器（无论是保持型还是非保持型）都将被清零，而且数据块的当前值被装载存储器的原始值覆盖。暖启动方式启动时，过程映像区和不保持的标志存储器、定时器及计数器被清零，保持的标志存储器、定时器和计数器以及数据块的当前值保持原状态。一般 S7-300PLC 都采用此种启动方式。热启动方式启动时，所有数据（无论是保持型和非保持型）都将保持原状态，然后程序从断点处开始执行。只有 S7-400 PLC 具有热启动功能。

“监视时间”选项用于设置相关项目的监控时间。

“来自模块的完成消息”用于设置电源接通后，CPU 等待所有被组态的模块发出“完成信息”的时间。“参数传递到模块的时间”用于将参数传递到模块的最长时间。

（2）扫描周期/时钟存储器的参数设置

扫描周期/时钟存储器的参数通过“周期/时钟存储器”选项卡设置，如图 3-22 所示。

“扫描周期监视时间”选项用于设置循环扫描时间，以“毫秒”为单位，默认值为 150ms，当实际扫描时间大于设定值时，CPU 进入 STOP 模式。

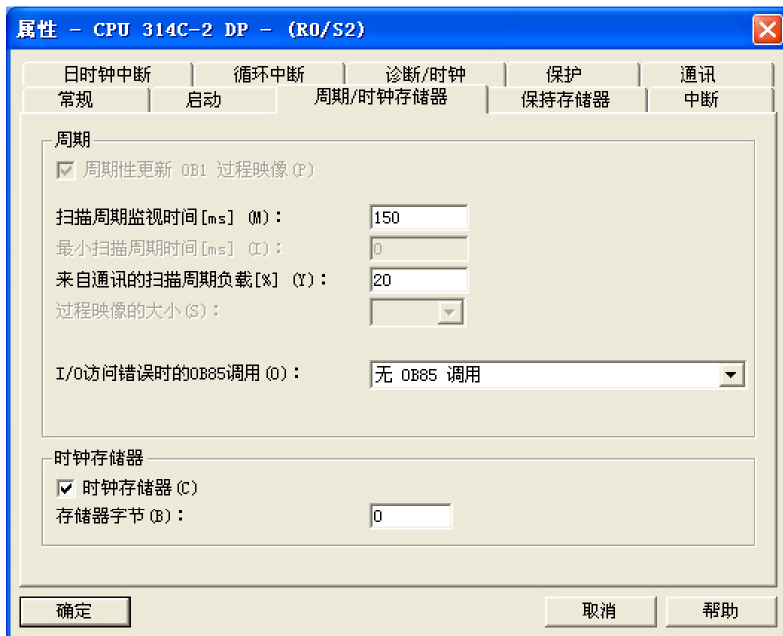


图 3-22　CPU 的属性窗口－周期/时钟存储器参数设置

时钟存储器用于设置时钟存储器的字节地址。S7-300/400 PLC 提供了一些不同频率的、占空比为 1:1 的方波脉冲信号给用户程序使用，这些方波信号存储在一个字节的时钟存储器



中，该字节默认为 MB0（此地址允许修改），需要将前面的复选框选中才能激活，该字节的每一位对应一种频率时钟脉冲信号，见表 3-1。

表 3-1　时钟存储器位与时钟脉冲周期和频率对应表

位	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
周期/s	2	1.6	1	0.8	0.5	0.4	0.2	0.1
频率/Hz	0.5	0.625	1	1.25	2	2.5	5	10

例如，图 3-22 中的时钟存储器为 MB0，所以 M0.5 的频率为 1Hz，可以用在明暗闪烁频率为 1Hz 报警灯中。

（3）系统诊断与时钟的参数设置

系统诊断与时钟的参数可以通过属性窗口的“诊断/时钟”选项卡设置，如图 3-23 所示。



图 3-23　CPU 的属性窗口—诊断/时钟参数设置

通过系统诊断可以发现程序的错误、模块的故障以及传感器和执行器的故障等。故障诊断可以通过选择“报告 STOP 模式原因”等选项设置。

（4）保持存储器的参数设置

保持存储器的参数可以通过属性窗口的“保持存储器”选项卡设置，如图 3-24 所示。

所谓保持存储器就是电源掉电或 CPU 从 RUN 进入 STOP 模式后内容保持不变的存储区。安装了后备电池的 S7 系列 PLC，用户程序中的数据块总是保存在保持存储区，没有后备电池的 PLC 可以在数据块中设置保持区域。

图 3-24 中默认的保持存储区为 MB0~MB15，C0~C7。



图 3-24　CPU 的属性窗口－保持存储器参数设置

（5）口令保护和运行方式的设置

口令保护和运行方式可以通过属性窗口的“保护”选项卡设置，如图 3-25 所示。

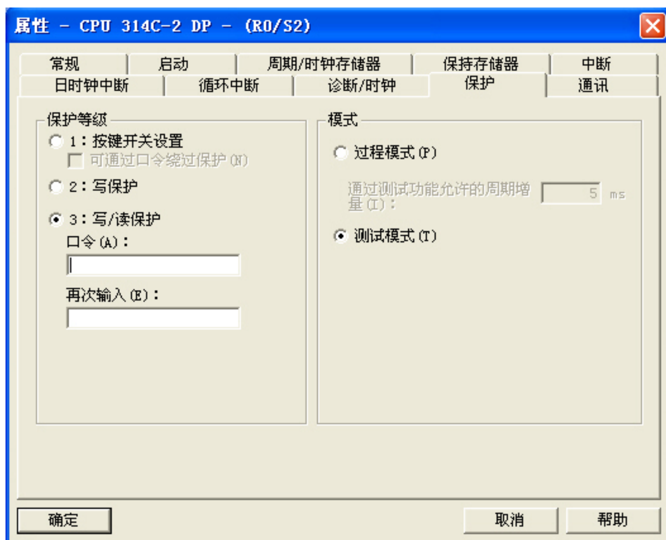


图 3-25　CPU 的属性窗口－口令保护和运行方式的设置

在 S7-300/400 PLC 中，使用口令保护功能可以使 CPU 中的程序和数据，有效防止对控制过程进行的可能的人为干扰。设置完成后将其下载到 CPU 模块中。

“保护”选项卡中有 3 个保护级别：保护级别 1，不需要设置口令；保护级别 2，只能进行读操作，而不能进行写操作；保护级别 3，只有拥有授权，才能进行读/写操作。口令的设置很容易，在图 3-25 的“口令”方框中输入字母或者数字，单击“确定”按钮即可。

（6）中断参数的设置

如图 3-26 所示的“中断”选项卡中，可以设置硬件中断、时间延迟中断和异步错误中



断的优先级。默认情况下，所有的硬件中断都由 OB40 来处理，用户可以通过设置优先级屏蔽中断。



图 3-26　CPU 的属性窗口—中断参数的设置

（7）循环中断参数的设置

如图 3-27 所示的“循环中断”选项卡中，可以设置循环执行组织块 OB30~OB38 的参数，这些参数包括中断优先级、以毫秒为单位的执行时间间隔和相位偏移量。例如，图 3-27 中每 100ms 执行一次组织块 OB35 中的程序。



图 3-27　CPU 的属性窗口—循环中断参数的设置

2．数字量输入/输出模块的参数设置

数字量输入/输出模块的参数分为动态参数和静态参数，在 CPU 处于 STOP 模式时，通过 STEP 7 的硬件组态，可以设置两种参数，参数设置完成后，应将参数下载到 CPU 中，这样当 CPU 从 STOP 转为 RUN 模式时，CPU 将参数自动传送到每个模块。

用户程序运行时，可以通过系统 SFC 调用修改动态参数。但当 CPU 从 RUN 模式进入 STOP 又返回 RUN 模式后，PLC 的 CPU 将重新传送 STEP 7 设置的参数到模块中，动态设

置的参数丢失。

（1）数字输入模块的参数设置

打开 HW Config 硬件组态界面，双击数字量输入模块所在的行，便可弹出属性窗口，选择某一选项卡，便可对其相应的属性进行设置。

在“地址”选项卡中可以设置数字量输入模块的起始字节的地址。若要修改起始地址，先要把“系统默认”前的复选框上的“√”去掉，再在“开始”后的框中输入新的起始地址即可，最后单击“确定”按钮，如图 3-28 所示。

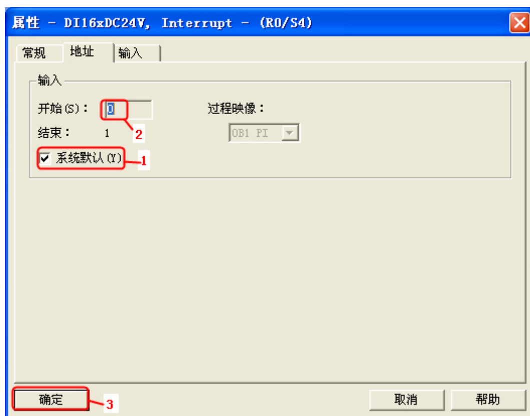


图 3-28　数字量输入模块—地址参数的设置

对于有中断功能的数字量输入模块，还有“输入”选项卡。可以通过复选框，选择是否允许产生“诊断中断”和“硬件中断”。如图 3-29 所示，先激活“硬件中断”，再激活输入点 0 和 1 位的“上升沿”，最后单击“确定”按钮，这样设置的含义是：当这个数字量输入模块的第 0 或者 1 位有上升沿时，触发硬件中断，CPU 将调用 OB40 进行处理。

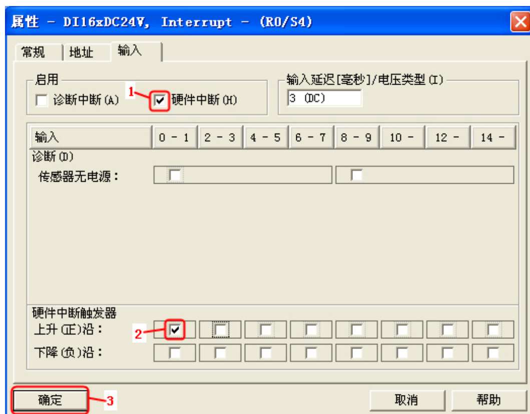


图 3-29　数字量输入模块—输入参数的设置

（2）数字输出模块的参数设置

在“地址”选项卡中可以设置数字量输出模块的输出起始地址，设置方法和数字量输入模块的类似。注意，当重新设置的地址若已经占用，则修改是不成功的。



有些输出模块有诊断中断、输出强制值功能，可以在“输出”中设置。在该选项卡中单击复选框可以设置是否允许产生诊断中断。“对 CPU STOP 模式的响应”下拉列表可以选择 CPU 进入 STOP 模式时，模块对各输出点的处理方式。选择“保持前一个有效值”，则 CPU 进入 STOP 模式后，模块将保持最后的输出值；而选择“替换值”，CPU 进入 STOP 模式后，可以使各点输出一个固定值，该值由“替换值 1”选项的复选框决定。如图 3-30 所示，替换值的所有的复选框都激活，所以当 CPU 进入 STOP 模式后，所有选中的输出点都为“1”。



图 3-30　数字量输出模块—输出参数的设置

3．模拟量输入/输出模块的参数设置

（1）模拟量输入模块的参数设置

模拟量输入模块的地址可以在“地址”选项卡中修改，方法与数字量模块的类似。模拟量模块“输入”选项卡如图 3-31 所示。



图 3-31　模拟量输入模块—输入参数的设置

如果已激活“诊断中断”复选框并发生诊断事件，则相应信息会输入到模块的诊断数据区。然后，模块会触发诊断中断，S7-CPU 会调用诊断中断模块 OB82（将在后面的章节讲解）。如果已激活“超出限制时硬件中断”复选框，输入值超出“上限值”和“下限值”定





义的范围，模块会触发硬件中断。

在“输入”选项卡中，还可以对模块的每一个通道的测量类型和测量量程。单击通道组的“量程型号”输入框，在弹出的菜单中选择测量种类，其中“E”表示测量电压信号、“4DMU” 表示四线式传感器的电流信号测量、“2DMU”表示两线式传感器的电流信号测量、“R-4L”表示 4 导线式电阻测量、“RT”表示热敏电阻测量温度信号、“TC-I”表示热电偶测量温度，如图 3-32 所示。单击“测量范围”输入框，弹出测量范围菜单（供选择，仅以测量电压信号为例说明），若输入电压信号的范围是-5V～+5V，则选择“+/-5V”选项，如图 3-33 所示。

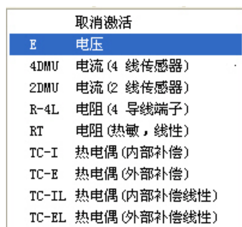


图 3-32　测量类型的设置

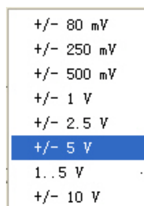


图 3-33　测量范围（电压）的设置

（2）模拟量输出模块的参数设置

模拟量输出模块的地址可以在“地址”选项卡中修改，方法与数字量模块的类似。模拟量模块“输出”选项卡如图 3-34 所示。

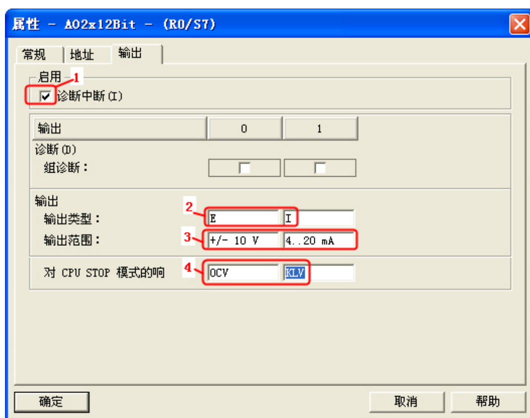


图 3-34　模拟量输出模块—输出参数的设置

可以设置各个通道是否允许中断，含义与模拟量输入模块相同。

模拟量的“输出类型”有 3 种“电流”、“电压”和“取消”，只要单击通道“测量类型”的方框，在弹出的菜单中选取即可。“电流”的含义是：输出的模拟信号是电流信号，“电压”的含义是：输出的模拟信号是电压信号，“取消”的含义是：不使用此通道。

“输出”范围的含义是输出模拟信号的范围，选择的方法是只要单击通道“输出范围”的方框，在弹出的菜单中选取即可。

对 CPU STOP 模式的响应有 3 个选项，其中“OCV”的含义是不输出电流电压，“KLV”的含义是保持前一个输出的电流电压值，“SV”的含义是采用替代值。



3.3.3　硬件的更新和GSD文件安装

1．硬件的更新

西门子的硬件更新比较快，每一个 STEP 7 版本都不可能包含未来的硬件。从 STEP 7 5.2 开始，STEP 7 提供了硬件更新功能。在组态硬件时，有时无法找到需要配置的硬件（一般以订货号为准），这就需要更新目录中的硬件。

硬件的更新有两种方法：一种是从磁盘上复制更新，另一种是从互联网上更新。从互联网上更新比较方便。方法如下：

首先打开 STEP 7 的硬件组态界面，单击菜单栏的“选项”→“安装 HW 更新”，如图 3-35 所示，弹出要求选择的更新的路径，如图 3-36 所示，选择“从 Internet 下载”更新，再单击“执行”按钮，自动搜索需要下载的硬件，如图 3-37 所示。



图 3-35　打开“硬件更新”的路径

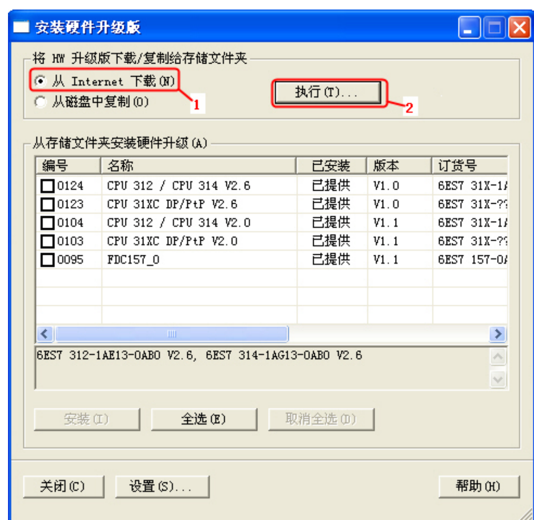


图 3-36　从 Internet 上更新

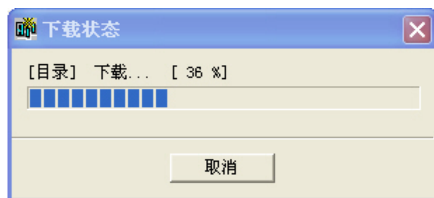


图 3-37　下载过程

搜索完毕后，弹出要求下载的硬件列表，选择需要下载的硬件，如图 3-38 所示，再单击“下载”按钮，下载完成后，自动弹出已经下载的硬件列表，如图 3-39 所示。选择需要安装的硬件并单击“安装”按钮，按照提示安装，安装时，STEP 7 要关闭。安装完成后重新打开 STEP 7，重新安装的硬件就可以使用了。



图 3-38　选择需要下载的硬件



图 3-39　选择需要安装的硬件

2．安装GSD文件

（1）GSD 文件简介

PROFIBUS 设备具有不同的性能特点，为达到 PROFIBUS 简单的即插即用配置。PROFIBUS 设备的特性均在电子设备数据库文件（GSD）中具体说明。标准化的 GSD 数据将通信扩大到操作员控制级。使用基于 GSD 的组态工具可将不同厂商生产的设备集成在同一总线系统中，既简单，又对用户友好。

（2）安装 GSD 文件



例如，安装了 STEP 7 V5.4 是不能组态 EM277 的（S7-300/400 PLC 与 S7-200 PLC 进行 PROFIBUS 通信需要组态此模块），因为 STEP 7 V5.4 中没有 SIEM089D.GSD 文件，因此必须安装此文件才能组态 EM277 模块。安装 SIEM089D.GSD 文件的方法如下：

首先打开 STEP 7 的硬件组态界面，单击菜单栏的“选项”→“安装 GSD 文件”，如图 3-40 所示，弹出要求选择安装 GSD 文件的路径，如图 3-41 所示，先选择 GSD 文件存放的目录，本例为“D:\Software\GSD”，再选择要安装的 GSD 文件，本例为“SIEM089D.GSD”，最后单击“安装”按钮即可。



图 3-40　打开“安装 GSD 文件”的路径



图 3-41　选择需要安装的硬件

3.4　下载和上传

3.4.1　下载

STEP 7 可以把用户的组态信息（SDB－系统数据）和程序下载到 CPU 中。下载的常用方法有：在项目管理器画面、在具体的程序或组态画面和离线/在线画面中进行下载。

在下载过程中，STEP 7 会提示用户处理相关信息，如：是否要删除模块中的系统数据，并离线系统数据替代，OB1 已经存在是否覆盖，是否停止 CPU 等，用户应该按照这些提示做出选择，完成下载任务。

当下载时把 CPU 面板的模式开关切换到“RUN”或“RUN-P”模式，下载信息包含硬件组态信息或网络组态信息等系统数据时，会提示需要切换到停止状态。

（1）在项目管理器中下载

在管理器中下载既可以下载整个站，也可以只下载一部分，例如只下载一个块。下面先介绍下载整个站的步骤。

首先打开 STEP 7 的项目管理器界面，单击菜单栏的“选项”→“设置 PG/PC 接口”，如图 3-42 所示，弹出“设置 PG/PC 接口”界面，如图 3-43 所示，选择“PC Adapter(MPI)”，单击“属性”按钮。

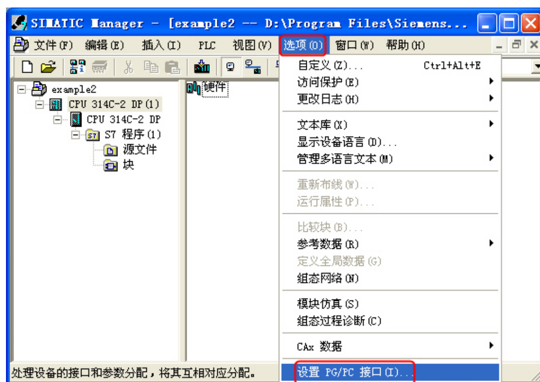


图 3-42　打开“设置 PG/PC 接口”界面

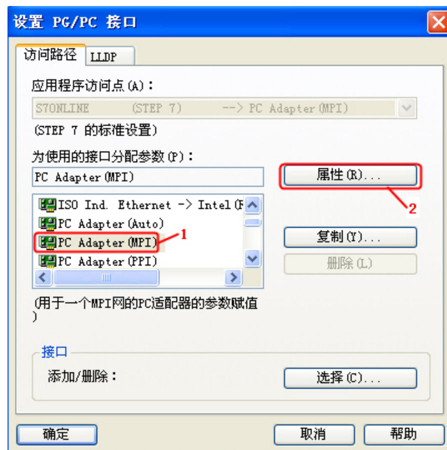


图 3-43　“设置 PG/PC 接口”界面

【关键点】下载软硬件的方法很多，可以用 MPI、PROFIBUS、S7、ISO 和 TCP 等通信协议下载，可以采用 MPI 适配器、以太网和 CP5611 卡（还有 CP5613、CP5511 等）传输介质下载，本例仅使用 MPI 协议，用 MPI 适配器下载程序，还有很多其他下载程序的方法。



在“MPI”选项卡中，设置编程器（或者 PC）的地址，本例为“1”，在设定传输率为“187.5Kbps”，如图 3-44 所示。

【关键点】编程器（或者 PC）的地址是唯一的，一般选用默认值“1”，传输率一般也选用默认值“187.5Kbps”。



图 3-44　通信参数设置

在“本地连接”选项卡中，设置编程器（或者 PC）的通信接口，本例为“USB”，再单击“确定”按钮即可，如图 3-45 所示。

【关键点】MPI 适配器一端与编程器相连，通常是 USB 接口或者 RS-232C 接口，另一端与 PLC 相连，通常是 RS-485 接口。若 MPI 适配器上有 USB 接口，则必须在 STEP 7 中安装驱动程序，此程序可以在西门子的官方网站（http://www.ad.siemens.com.cn/）下载。



图 3-45　设置通信接口

回到如图 3-43 所示的界面，单击“确定”按钮，弹出如图 3-46 所示界面，单击“确定”按钮。

【关键点】若上一次使用的通信协议与这次相同，则不会有此界面弹出。

先选取站点“S7-300”，再单击“下载”按钮，如图 3-47 所示，程序和硬件组态开始下载，若组态模块的信息和实际目标模块的信息不同时，弹出如图 3-48 所示界面，若订货号等有报警（即出现一个黄色三角形，三角形中有一个“！”），则必须重新组态。

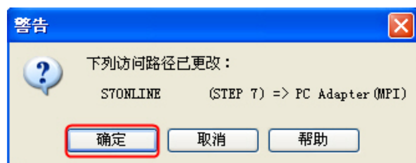


图 3-46　更改访问路径

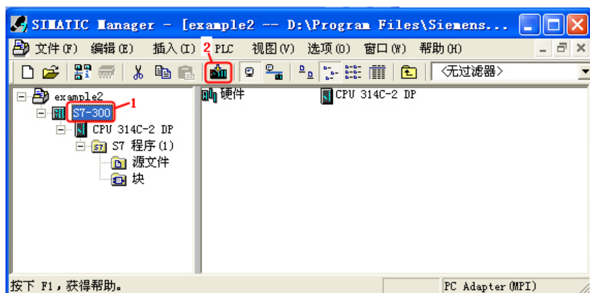


图 3-47　项目管理器画面



图 3-48　组态模块的信息和实际目标模块的信息对照

如果 PLC 中已经有程序或者硬件组态，则会弹出如图 3-49 所示界面，含义为是否将 CPU 中的“OB1”用编程器中的替代（或者说覆盖），单击“全部”按钮，表示全部覆盖。当所有的程序下载完成后，弹出如图 3-50 所示界面，提示是否重新启动 CPU，单击“是”按钮，则重启 CPU。

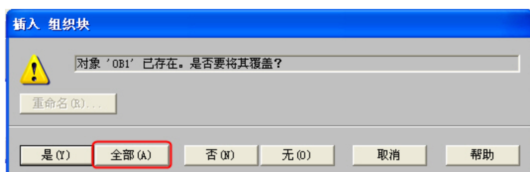


图 3-49　是否“覆盖 OB1”信息

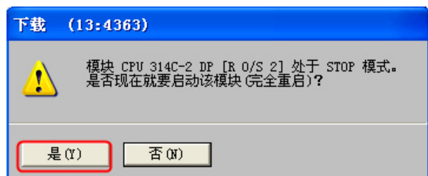


图 3-50　重启信息

（2）在具体的程序或者组态中下载

如果硬件组态已经下载到 CPU，则接下来只要下载程序即可，没有必要每次下载时，都将整个站全部下载。

首先打开 STEP 7 的程序编辑器界面，再单击“下载”按钮即可，如图 3-51 所示。

【关键点】程序有语法错误是不能下载的。

如果硬件组态和程序已经下载到 CPU，只对硬件组态做了修改，没有必要每次下载时，都将整个站全部下载，只下载硬件组态即可。

首先打开 STEP 7 的硬件组态界面，再单击“下载”按钮即可，如图 3-52 所示。

【关键点】硬件组态中有错误是不能下载的。



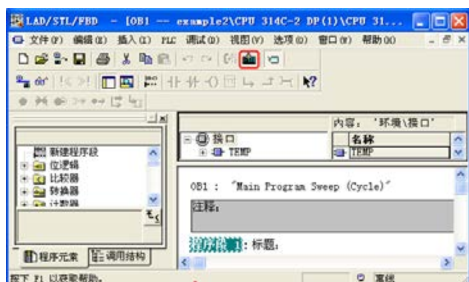


图 3-51　在“程序编辑器”中下载程序



图 3-52　在“硬件组态界面”中硬件组态

3.4.2　上传

STEP 7 可以把 CPU 中的组态信息和程序上传到用户的项目中。上传常用的方法有在项目管理器界面、在硬件组态界面和在线/离线界面中进行上传。

1．在项目管理器中上传

在项目管理中新建一个空项目“Upload”，单击菜单栏“PLC”→“将站点上传到 PG”，如图 3-53 所示，弹出“选择节点地址”对话框，先选择目标站点为“本地”，单击“更新”按钮，如图 3-54 所示，单击“确定”按钮，上传开始，上传过程如图 3-55 所示。

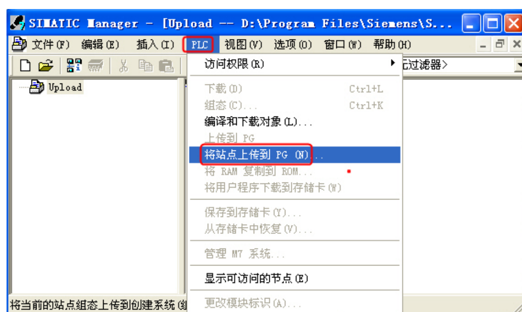


图 3-53　在项目管理中上传的路径

【关键点】用这种方法上传的是站点，包括硬件组态信息和程序。

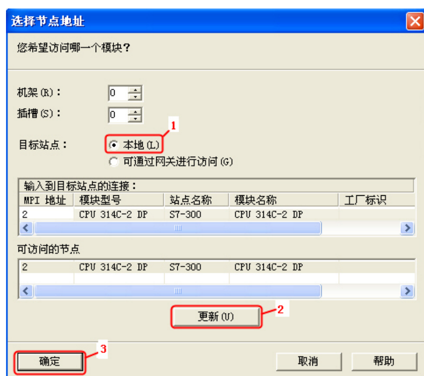


图 3-54　“选择节点地址”对话框

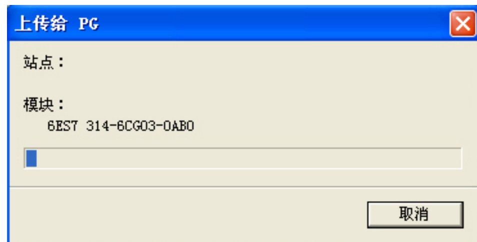


图 3-55　上传过程

2．在组态界面中上传

在组态界面中上传，先打开 STEP 7 的硬件组态界面，如图 3-56 所示，单击工具栏的“上传”按钮，弹出选择目标项目对话框，选择目标项目，本例为“Upload”，如图 3-57 所示，弹出“选择节点地址”对话框，先选择目标站点为“本地”，单击“更新”按钮，如图 3-58 所示，单击“确定”按钮，上传开始，上传过程如图 3-55 所示。

【关键点】用这种方法上传的信息只有硬件组态信息。

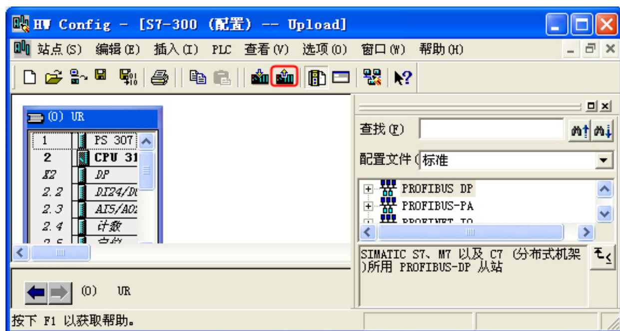


图 3-56 上传

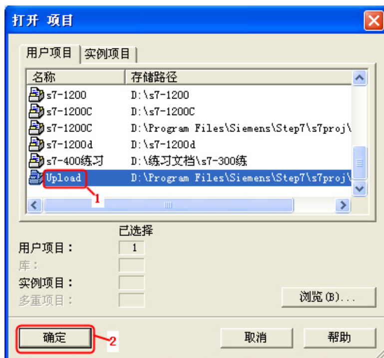


图 3-57 选择目标项目

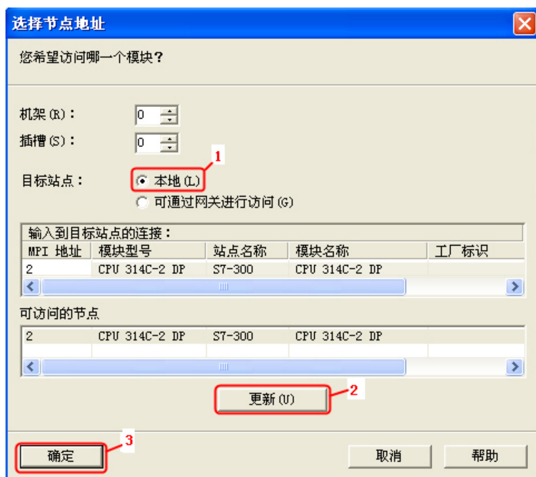


图 3-58 “选择节点地址”对话框

3.5 软件编程

不管什么 PLC 工程，编写程序总是必需的，编写程序在硬件组态完成后，主程序编写在 OB1 组织块中，其他程序如时间循环中断程序可编写在 OB35 中。以下介绍一个最简单的程序的输入和编译过程。

1．打开程序编译器。

首先在管理器界面，选中并双击“OB1”组织块，打开程序编辑器界面，如图 3-59 所示。



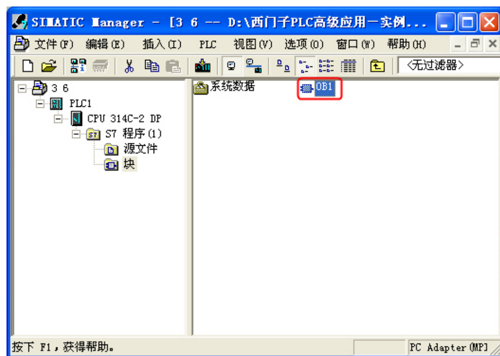


图 3-59　管理器界面

2．输入程序。

选中如图 3-60 所示的“1”处，双击“常开触点”┤├，常开触点自动跳到“1”处，再用同样的方法双击“线圈”-()，线圈跳到常开触点的后面，如图 3-61 所示。

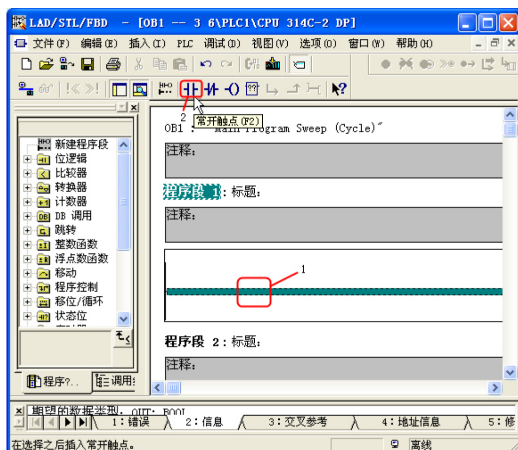


图 3-60　输入程序（1）

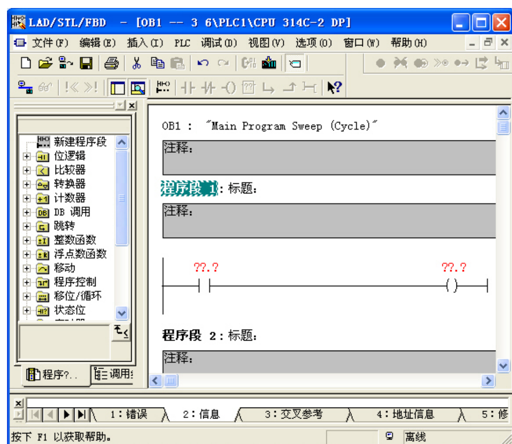


图 3-61　输入程序（2）





单击“常开触点”上的红色“??.?”，弹出方框，在方框中输入“I0.0”，按〈Enter〉键即可。同理，输入“Q0.0”，按〈Enter〉键，如图 3-62 所示。

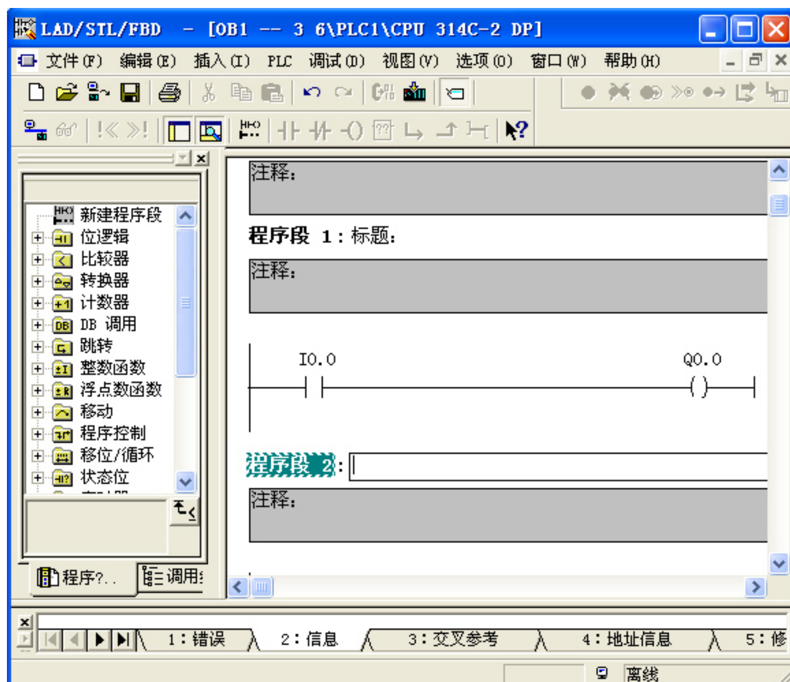


图 3-62　输入程序（3）

3．保存程序。

单击工具栏上的按钮即可。

3.6　打印和归档

一个完整的项目工程包含文字、图表和程序文件。打印的目的就是进行纸面上的交流和存档，项目归档是电子方面的交流和存档。

3.6.1　打印

打印的操作步骤如下：

1）打开相应的项目对象，在屏幕上显示要打印的信息。

2）在应用程序窗口中，使用菜单栏命令“文件”→“打印”，打开打印界面。

3）可以在对话框中更改打印选项（例如打印机、打印范围和打印份数等），也可以将程序等生成 mdi 或者 pdf 格式的文档。

以下介绍生成 mdi 格式的文档步骤：

在程序编辑器中，使用菜单栏命令“文件”→“打印”，打开“打印”对话框，如图 3-63 所示，打印机名称选择“Microsoft Office Document Image Writer”，单击“确定”按钮，生成 mdi 格式的文档如图 3-64 所示。



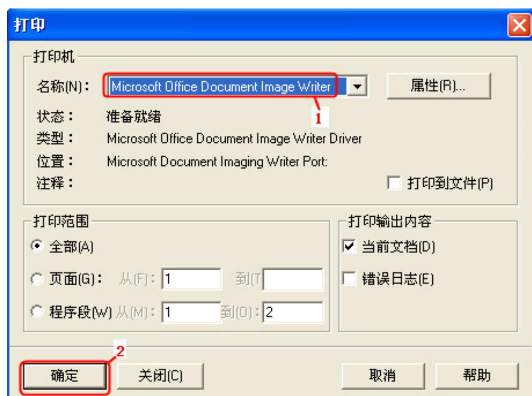


图 3-63 “打印”对话框

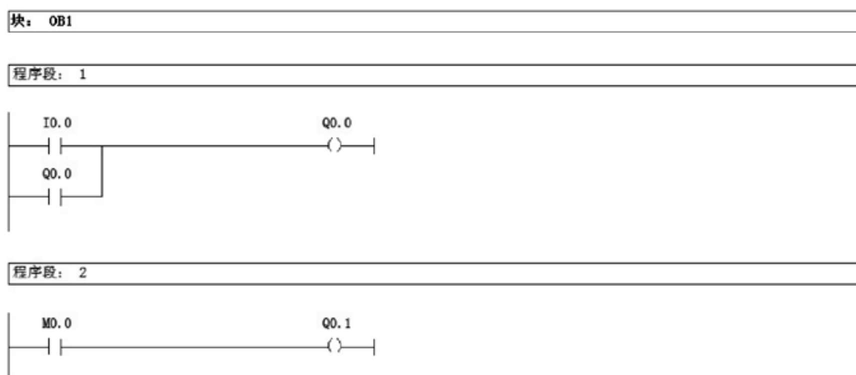


图 3-64 程序生成 mdi 格式的文档例子

3.6.2 归档

项目归档的目的是把整个项目的文档压缩到一个压缩文件中，以方便备份和转移。当需要使用时，使用解压的方式恢复为原来项目的文档。归档的具体步骤如下：

打开项目资源管理器，单击菜单栏的“文件”→“归档”，如图 3-65 所示，弹出查找将要归档的项目名称，如图 3-66 所示，单击“确定”按钮，弹出选择归档的路径，如图 3-67 所示，单击“保存”按钮，生成一个扩展名为“.zip”的压缩文件。

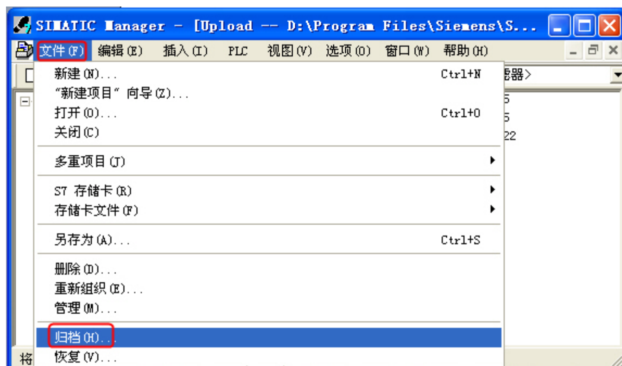


图 3-65 归档的路径





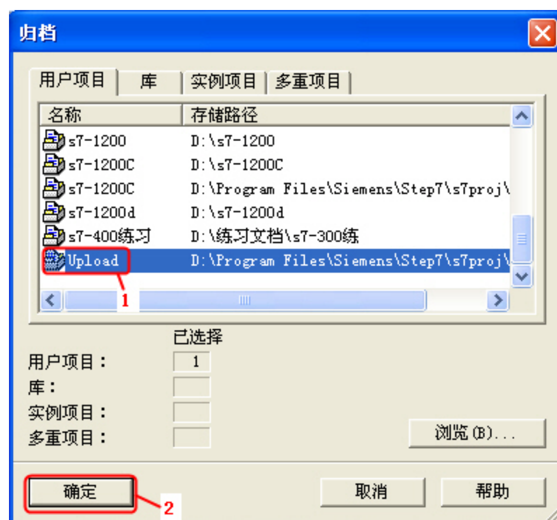


图 3-66　选择将要归档的项目名称

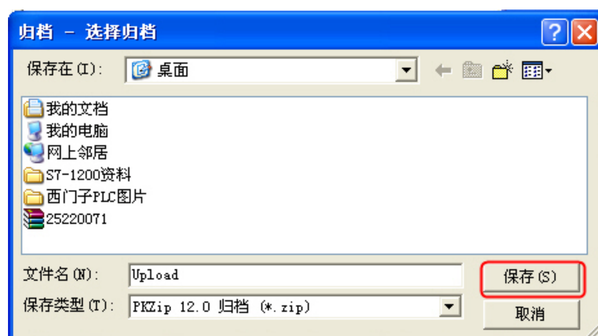


图 3-67　选择归档的路径

3.7　用STEP 7 V5.4 建立一个完整的项目

以下创建一个项目，用一台 CPU 314C-2DP 控制一盏灯的开和关。

1．先设计原理图，如图 3-68 所示，并按照原理图接线

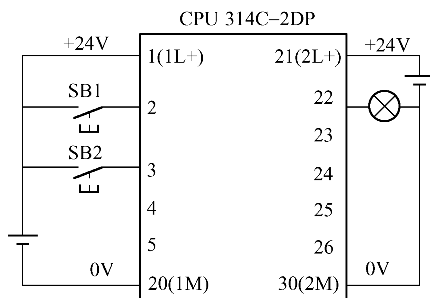


图 3-68　原理图



2．打开软件STEP 7，新建项目，并进行硬件组态

1）打开软件 STEP 7。双击 SIMATIC Manager 图标，打开 STEP 7 软件。

2）新建项目。单击工具栏上的按钮，弹出界面如图 3-69 所示，在“名称”中输入“example”，单击“确定”按钮。

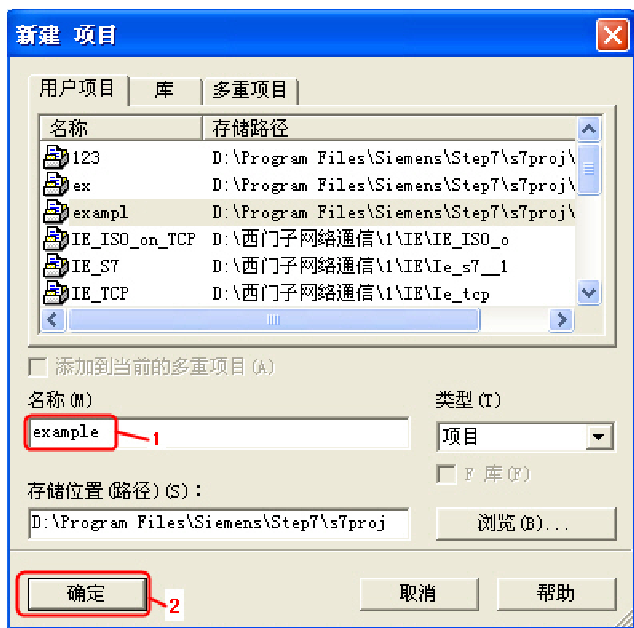


图 3-69　新建项目

3）插入站点。选中管理器中的项目名称“example”，再单击菜单栏中的“插入”→“站点”→“SIMATIC 300 站点”，如图 3-70 所示。

4）启动硬件组态界面。单击项目名称“example”左侧的“＋”，选定“SIMATIC 300（1）”→“双击硬件”，之后弹出硬件组态界面，如图 3-71 所示。



图 3-70　插入站点

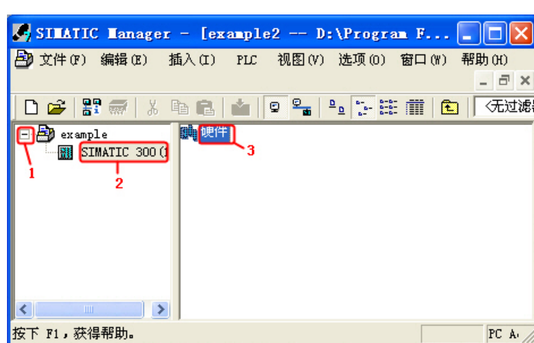


图 3-71　启动硬件组态界面

5）插入机架。在硬件组态界面中，单击机架“Rail”，机架自动弹出，如图 3-72 所示。

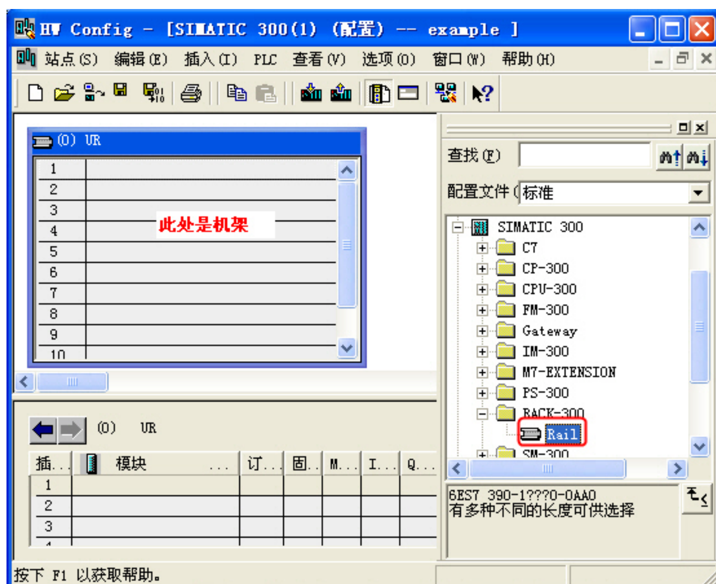


图 3-72　插入机架

6）插入电源模块。在硬件组态界面中，先选中“1”处，再双击“PS 307 5A”，电源模块自动安装到 1 号槽位，如图 3-73 所示。也可以用鼠标的左键选中“PS 307 5A”，并按住不放，将电源模块拖至 1 号槽位。

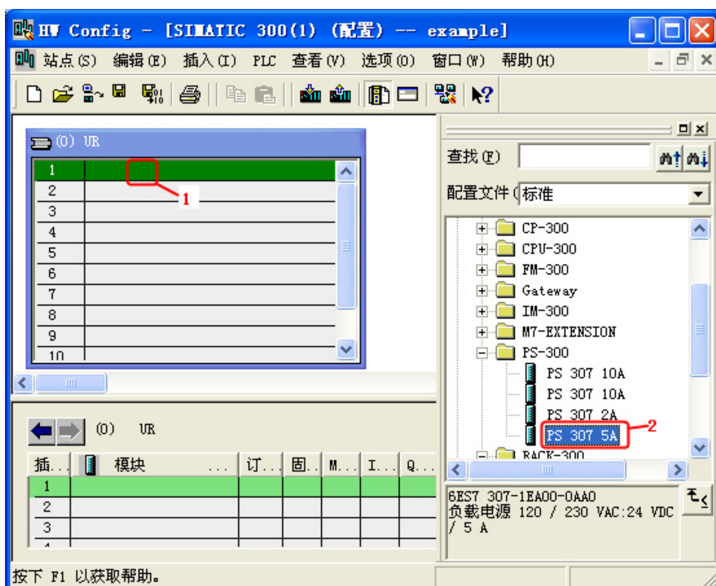


图 3-73　插入电源模块

7）插入 CPU 模块。在硬件组态界面中，用鼠标的左键选中“V2.6”，并按住不放，将 CPU 模块拖至 2 号槽位，如图 3-74 所示。也可以先选中“1”处，再双击“CPU 314-2 DP”下的“V2.6”，CPU 模块自动安装到 2 号槽位。



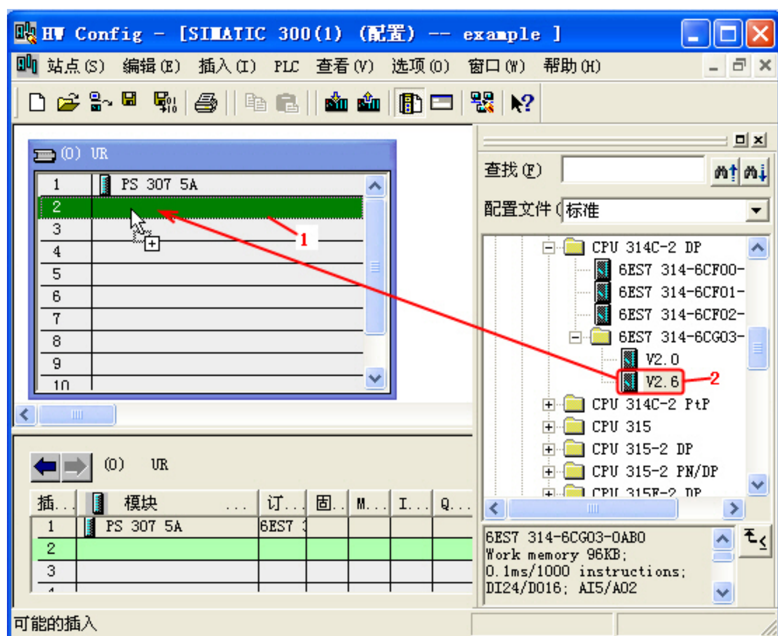


图 3-74　插入 CPU 模块（用拖放方法）

8）保存和编译组态。至此一个简单的项目已经组态完成，只要单击工具栏上的“保存和编译”按钮即可，如图 3-75 所示。

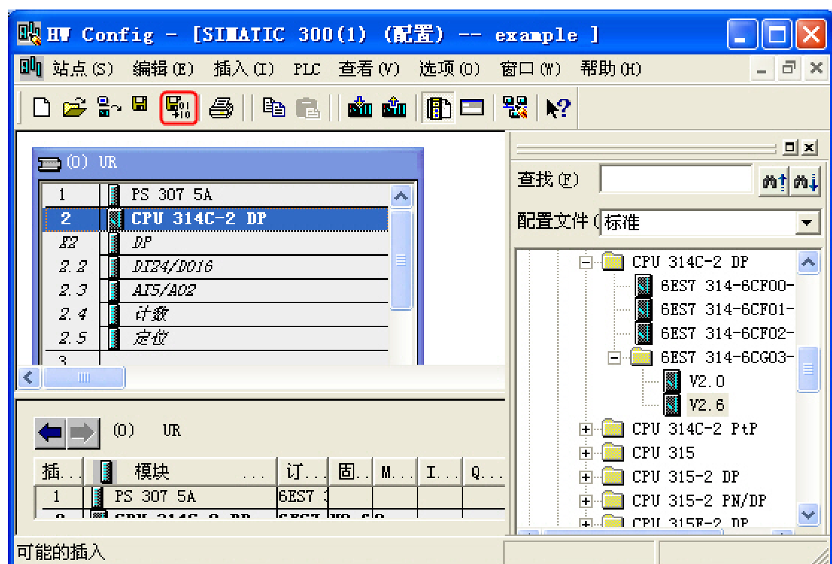


图 3-75　保存和编译组态

3．编写程序

打开程序编辑器，在编辑器中编写程序，如图 3-76 所示，再单击工具栏上的“保存”按钮，保存程序。

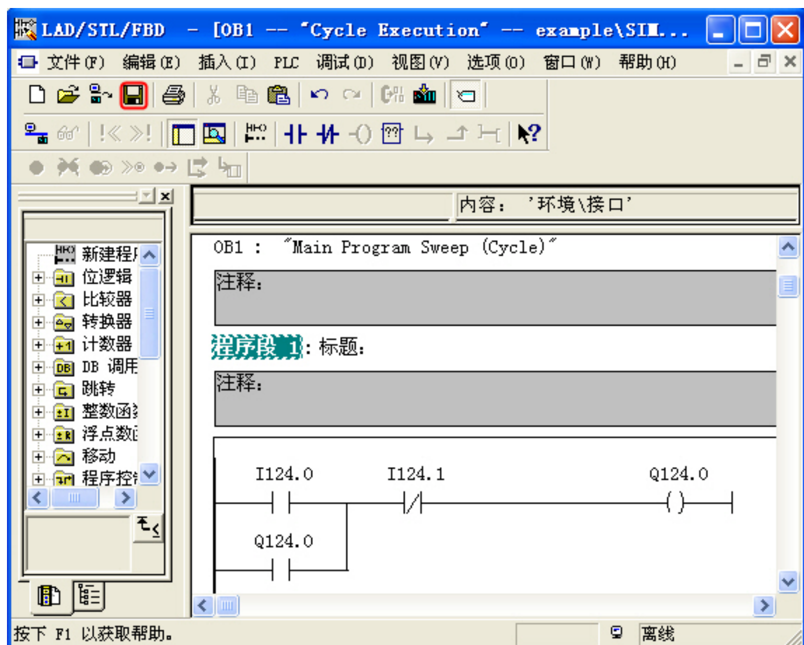


图 3-76　程序

【关键点】CPU 314C-2DP 的集成数字 I/O 的默认起始地址是 IB124 和 QB124，当然这个地址是可以修改的，如修改成 IB0 和 QB0。

4．下载硬件组态和程序

先把 MPI 适配器将 CPU 314C-2DP 与编程器（PG）相连，再通电。

打开项目管理器，选中站点“SIMATIC 300 站点”，再单击工具栏上的“下载”按钮 ，如图 3-77 所示，整个站点（含程序和硬件组态）下载到 CPU 中。当合上按钮 SB1 时，灯亮，当合上按钮 SB2 时，灯灭。

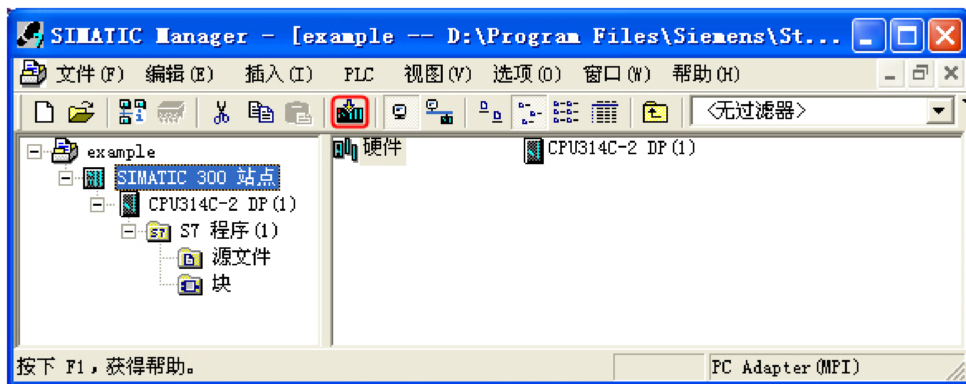


图 3-77　站点下载

【关键点】CPU 314C-2DP 在通电状态时，不要直接拔插 MPI 适配器，这样操作可能会



导致 CPU 的 MPI 接口烧毁，应该先断开 CPU 电源，再拔插 MPI 适配器。

3.8 使用帮助

STEP 7 软件有强大的帮助功能，早期的版本只有英文版，因此帮助全部是英文，而现在，只要安装中文版的 STEP 7 软件，则帮助以中文为主，少数模块（例如后安装的 FM355）的帮助仍然是英文。学会使用帮助是非常重要的，因为 STEP 7 软件的帮助信息十分全面，是学习和使用 S7-300/400 PLC 的必备工具。

3.8.1 查找关键字或者功能

在工作或者学习时，可以利用“关键字”搜索功能查找帮助信息。以下用一个例子说明查找的方法。

先在项目管理器的菜单栏中，单击“帮助”→“目录”，此时弹出帮助信息界面，选中“搜索”选项卡，再在“键入要查找的关键字”输入框中，输入关键字，本例为“OB82”，单击“列出主题”，则有关“OB82”的信息全部显示出来，读者通过阅读这些信息，可了解“OB82”的用法，如图 3-78 所示。

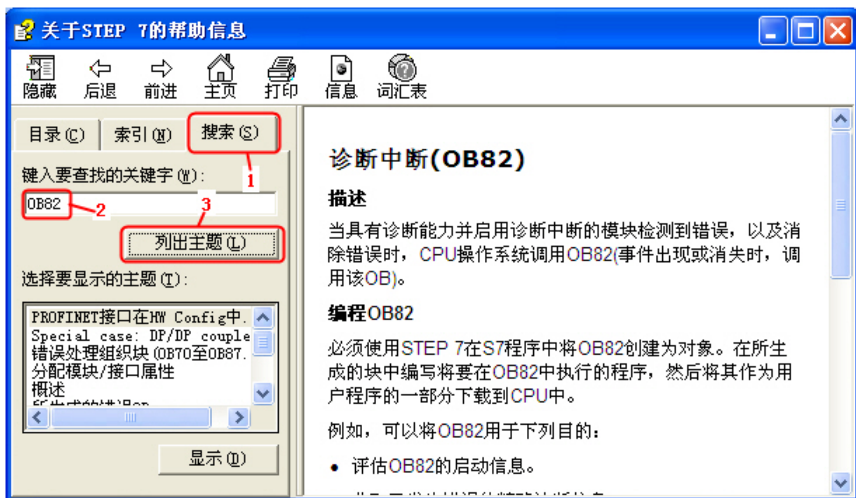


图 3-78 帮助—搜索

3.8.2 了解某个逻辑块FB/SFB/FC/SFC的功能及管脚的定义

STEP 7 中内置了很多逻辑块（FB/SFB/FC/SFC），在学习和应用时，不可避免会对某个逻辑块有疑问。解决的方法是：在程序编辑器中，先找到这个逻辑块，本例为“SFB8”，先选中“SFB8”，如图 3-79 所示，再按键盘上的〈F1〉键，弹出“SFB8”的帮助界面，如图 3-80 所示。也可在程序编辑器中，单击工具栏上的“空逻辑框”按钮 ，并键入具体逻辑块名（本例为“SFB8”），再按键盘上的〈F1〉键，弹出“SFB8”的帮助界面，如图 3-79 所示。

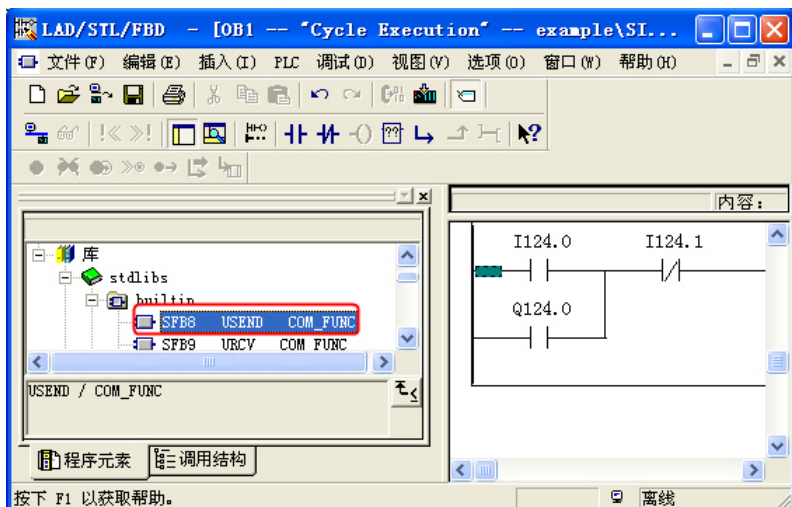


图 3-79　选中逻辑块

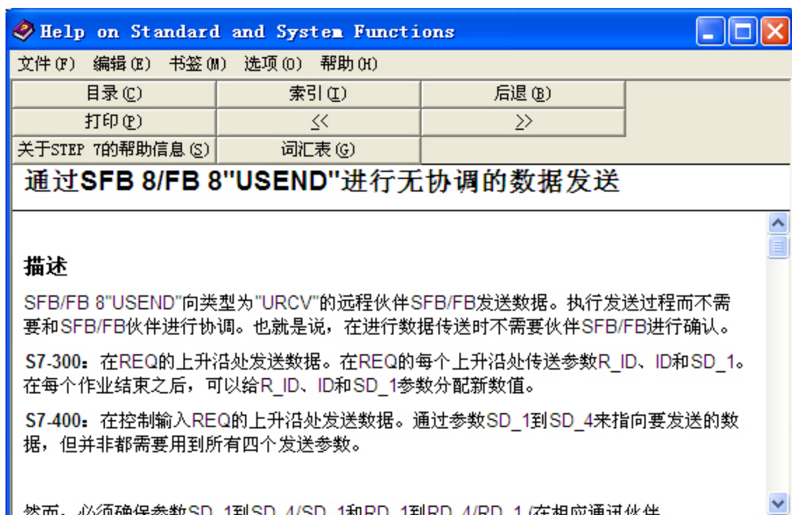


图 3-80　帮助界面

小结

重点难点总结

1．学会安装 STEP 7 软件，熟悉安装 STEP 7 软件的软硬件条件。

2．掌握 S7-300/400 PLC 的硬件组态、硬件的更新和安装 GDS 文件。

3．掌握上传和下载项目。

4．理解模块参数的含义，并会设置模块参数。

5．了解打印和归档。

6．掌握使用在线帮助。



7．用 STEP 7 V5.4 建立一个完整的项目。

习题

1．怎样新建一个新项目？

2．用于 S7-300/400 PLC 的编程语言有哪几种？

3．安装 STEP 7 软件要注意哪些事项？

4．若在组态硬件时，发现这个硬件在列表中找不到，应该怎样操作？

5．若在组态 PROFIBUS 网络时，发现找不到硬件，应该怎样操作？

6．向 S7-300/400 PLC 中上传和下载的步骤有哪些？

7．怎样设置 CPU 的扫描时间？

8．怎样设置 S7-300/400 PLC 密码？

9．怎样将 PLC 中的信息保存到 PC 的新建项目“Example”中。

10、怎样进行文件归档？

11．利用“在线帮助”，搜索系统功能块 SFC14/SFC15 的功能及其管脚定义。

12．若导轨上的 CPU 是 CPU 315（订货号：6ES7 315-1AF02-0AB0），输入模块是 SM321(订货号：6ES7 321-1BH01-0AA0)，输出模块是 SM322（订货号：6ES7 322-5HF00-0AB0），请利用以上硬件设计一个电气系统，控制一台三相异步电动机的起停，并创建 STEP 7 项目和编写控制程序。

13．怎样安装 GSD 文件？

14．拔插 MPI 适配器时，为什么要切断 CPU 的电源？

15．怎样复制和删除项目？

第4章

S7-300/400 PLC的编程语言

本章介绍 S7-300/400 PLC 的编程基础知识、指令系统、功能、功能块、系统功能、系统功能块、数据块、组织和中断。本章内容多，而且难点多，是入门的关键。

4.1 S7-300/400 PLC的编程基础知识

4.1.1 编程元件与数据类型

1．编程元件

S7-300/400 PLC 内部有许多编程元件。编程元件通常指的是硬件，是 PLC 内部具有一定功能的器件总称，这些器件由电子电路和寄存器及存储单元等组成。例如，输入继电器由输入电路和输入映像寄存器构成，输出继电器由输出电路和输出存储映像寄存器构成。定时器和计数器等由特定功能的寄存器构成。为了把这种继电器与传统的电气控制电路中的继电器区分开来，有时也称之为软继电器。这些软继电器的存储区及功能见表 4-1。

表 4-1　存储区及功能

地址存储区	范围	S7 符号	举例	功能描述
过程映像输入区	输入（位）	I	I0.0	扫描周期期间，CPU 从模块读取输入，并记录该区域中的值
	输入（字节）	IB	IB0	
	输入（字）	IW	IW0	
	输入（双字）	ID	ID0	
过程映像输出区	输出（位）	Q	Q 0.0	扫描周期期间，程序计算输出值并将它放入此区域，扫描结束时，CPU 发送计算输出值到输出模块
	输出（字节）	Q B	Q B0	
	输出（字）	Q W	Q W0	
	输出（双字）	Q D	Q D0	
位存储器	存储器（位）	M	M 0.0	用于存储程序的中间计算结果
	存储器（字节）	M B	M B0	
	存储器（字）	M W	M W0	
	存储器（双字）	M D	M D0	
定时器	定时器（T）	T	T3	为定时器提供存储空间
计数器	计数器（C）	C	C0	为计数器提供存储空间

（续）



地址存储区	范围	S7 符号	举例	功能描述
共享数据块	数据（位）	DBX	DBX 0.0	数据块用“OPN DB”打开，数据块包含的信息，可以被所有的逻辑块定义为通用（共享 DB）
	数据（字节）	DBB	DBB0	
	数据（字）	DBW	DBW0	
	数据（双字）	DID	DBD0	
背景数据块	数据（位）	DIX	DIX 0.0	数据块用“OPN DI”打开，数据块包含的信息，可以分配给特定的 FB 或者 SFB 逻辑块定义为背景数据块
	数据（字节）	DIB	DIB0	
	数据（字）	DIW	DIW0	
	数据（双字）	DID	DID0	
本地数据区	本地数据（位）	L	L 0.0	当块被执行时，此区域包含块的临时数据
	本地数据（字节）	LB	L B0	
	本地数据（字）	LW	LW0	
	本地数据（双字）	L D	LD0	
外部设备输入区	外部设备输入字节	PIB	PIB0	外围设备输入区允许直接访问中央和分布式的输入模块
	外部设备输入字	PIW	PIW0	
	外部设备输入双字	PID	PID0	
外部设备输出区	外部设备输出字节	PQ B	PQ B0	外围设备输出区允许直接访问中央和分布式的输入模块
	外部设备输出字	PQ W	PQ W0	
	外部设备输出双字	PQ D	PQ D0	
程序		FC/FB/ SFB/SFC	FC1/FB1/ SFB1/SFC2	FC 分用户、标准和系统功能 FB 分用户、标准和系统功能块

2．数据类型

数据是程序处理和控制的对象，在程序运行过程中，数据是通过变量来存储和传递的。变量有两个要素：名称和数据类型。对程序块或者数据块的变量声明时，都要包括这两个要素。

数据的类型决定了数据的属性，例如数据长度和取值范围等。STEP 7 中的数据类型分为 3 大类：基本数据类型、复杂数据类型和参数数据类型。

（1）基本数据类型

基本数据类型是根据 IEC1131-3（国际电工委员会指定的 PLC 编程语言标准）来定义的，每个基本数据类型具有固定的长度且不超过 32 位。

基本数据类型有 12 种，每一种数据类型都具备关键字、数据长度、取值范围和常数表等格式属性。STEP 7 的基本数据类型见表 4−2。

表 4−2　STEP 7 的基本数据类型

关键字	长度（位）	取值范围/格式示例	说明
Bool	1	True 或 False（1 或 0）	布尔变量
Byte	8	B#16#0~ B#16#FF	单字节数
Word	16	十六进制：W#16#0~ W#16#FFFF	字（双字节）
Dword	32	十六进制：（W#16#0~ W#16#FFFF_FFFF）	双字（四字节）

（续）

关键字	长度（位）	取值范围/格式示例	说明
Char	8	“a”、“B”等	ASCII 字符
Int	16	−32768~32767	16 位有符号整数
DInt	32	−L#2147483648~ L#2147483647	32 位有符号整数
Real	32	−3.402823E38~−1.175495E−38 +1.175495E−38~+3.402823E38	IEEE 浮点数
S5Time	16	S5T＃0H_0M_0S_0MS~ S5T＃2H_46M_30S_0MS	SIMATIC 时间格式
Time	32	−T＃24D_20H_31M_23S_648MS~ T＃24D_20H_31M_23S_648MS	IEC 时间格式
Date	16	D#1990−1−1~ D#2168−12−31	IEC 日期格式
Time_Of_Day	32	TOD#0:0:0~ TOD#23:59:59.999	24 小时时间格式

【关键点】常数可以用二进制、十进制和十六进制表示。为了阅读方便，当用二进制和十六进制表示时，可以在每 4 位之间加下划线，例如 W#16#FFFF_FFFF 和 W#16#FFFFFFFF 实际是相等的。

（2）复杂数据类型

复杂数据类型是一种由其他数据类型组合而成的，或者长度超过 32 位的数据类型，STEP 7 中的复杂数据类型共有 7 类。

1）Date_And_Time（日期时间类型）。其长度为 64 位（8 个字节），此数据类型以二进制编码的十进制的格式保存。取值范围是 DT#1990-1-1-0:0:0.0~ D#2089-12-31-59:59.999。

2）STRING（字符串）。其长度最多有 254 个字符的组（数据类型 CHAR）。为字符串保留的标准区域是 256 个字节长。这是保存 254 个字符和 2 个字节的标题所需要的空间。可以通过定义即将存储在字符串中的字符数目来减少字符串所需要的存储空间（例如：string[9]'Siemens'）。

3）ARRAY（数组类型）。定义一个数据类型（基本或复杂）的多维组群。例如：“ARRAY [1..2,1..3] OF INT”定义 2 × 3 的整数数组。使用下标（“[2,2]”）访问数组中存储的数据。最多可以定义 6 维数组。下标可以是任何整数（−32768 ～ 32767）。

4）STRUCT（结构类型）。该类型是由不同数据类型组成的复合型数据，通常用来定义一组相关数据。例如电动机的一组数据可以按照如下方式定义：

```
Motor: STRUCT
    Speed: INT
    Current: REAL
END_STRUCT
```

5）UDT （用户自定义数据类型）。UDT 是由不同数据类型组成的复合型数据，与 STRUCT 不同的是，UDT 是一个模版，可以用来定义其他的变量。它在 STEP 7 中以块的形式存储，称为 UDT 块。在 S7 的项目管理器中，先选中“块”，再单击菜单栏的“插入”→“S7 块”→“数据类型”，如图 4−1 所示，弹出数据类型对话框便可定义新的数据类型。



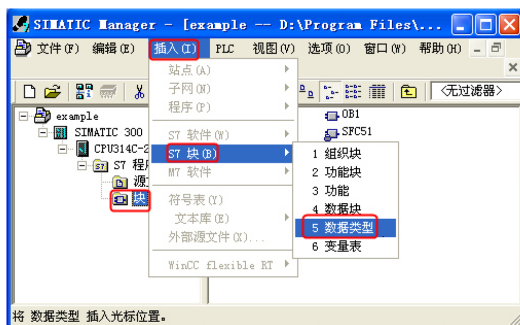


图 4-1 定义 UDT 块的路径

6）FB 和 SFB（功能块类型）。确定分配的实例数据块的结构，并允许在一个实例 DB 中传送数个 FB 调用的实例数据，在后面章节会重点讲解。

（3）参数数据类型

参数数据类型是一种用于 FC 或者 FB 的参数的数据类型。参数数据类型主要包括以下几种：

- Timer，Counter：定时器和计数器类型。
- BLOCK_FB，BLOCK_FC，BLOCK_DB，BLOCK_SDB：块类型。
- Pointer：6 字节指针类型，传递DB块号和数据地址。
- Any：10 字节指针数据类型，传递DB块号、数据地址、数据数量以及数据类型。

使用这些参数类型，可以把定时器、计数器、程序块、数据块以及一些不确定类型和长度的数据通过参数传递给 FC 和 FB。参数类型为程序提供了很强的灵活性，可以实现更通用的控制功能。

4.1.2 寻址方式

1. 绝对寻址

要访问一个变量，必须要找到它在存储空间的位置，这个过程就是寻址（Addressing）。在 STEP 7 中，使用地址如 I/O 信号、位内存、计数器、定时器、数据块和功能块都可以通过绝对寻址和符号寻址来访问。

绝对地址是由一个关键字和一个地址数据组成的。STEP 7 中常用的绝对地址关键字见表 4-3。

表 4-3 绝对地址关键字

关键字	说明	举例
I/IB/IW/ID	过程映像区输入信号	I0.0、IB1、IW2 和 ID2
Q/QB/QW/QD	过程映像区输出信号	Q0.0、QB1、QW2 和 QD2
PIB/PIW/PID	外部设备输入	PIB1、PIW2 和 PID2
PQB/PQW/PQD	外部设备输出	PQB1、PQW2 和 PQD2
M/MB/MW/MD	位存储区	M0.0、MB1、MW2 和 MD2
L/LB/LW/LD	本地数据堆栈区	L0.0、LB1、LW2 和 LD2
T	定时器	T3
C	计数器	C5
FC/FB/SFC/SFB	程序块	FC1/FB2/SFC3/SFB1
DB	数据块	DB1

【关键点】在 MD0 中，由 MB0、MB1、MB2 和 MB3 四个字节组成，MB0 是高字节，而 MB3 是低字节，字节、字和双字的起始地址如图 4-2 所示。

MB0	7　MB0　0			
MW0	15　MB0　8	7　MB1　0		
MD0	31　MB0　24	23　MB1　16	15　MB2　8	7　MB3　0
	最高有效字节			最低有效字节

图 4-2　字节、字和双字的起始地址

【例 4-1】　如果 MD0=1FH，那么，MB0、MB1、MB2 和 MB3 的数值是多少？

【解】

根据图 4-2：MB0＝0；MB1＝0；MB2＝0；MB3＝1FH。这点不同于三菱 PLC，读者要注意区分。

2．符号寻址

为变量指定符号名可以简化程序的编写和调试，增加程序的可读性。STEP 7 可以自动将符号地址转化成所需的绝对地址。访问 ARRAY、STRUCT、数据块、本地数据、逻辑块以及用户数据类型（UDT）时，优先选用符号寻址。使用符号寻址前，必须先将符号分配给绝对地址，才能以符号的形式应用他们。

STEP 7 中的符号分为全局符号和局域符号。全局符号是在整个 STEP 7 中可以使用的符号，而局域符号是在某个块中可以使用的符号。全局符号和局域符号的对比见表 4-4。

表 4-4　全局符号和局域符号的对比

	全局符号	局域符号
有效范围	在整个程序中有效，可以被所有的块使用，在所有的块中的含义是一样的，整个用户程序中是唯一的	只在定义的块中有效，相同的符号在不同的块中，可用于不同的目的
允许使用的字符	字母、数字及特殊字符，除 0x00，0xFF 及引号外的强调号；如用特殊符号，则必须在引号内	字母 数字 下划线（_）
使用独享	可以为下列对象定义全局变量： • I/O 信号（如 I、IB、Q、QB、QD 等） • 外部设备 I/O 信号（如 PIB、PQB、PQD 等） • 存储位（如 M、MB、MW、MD 等） • 定时器（T） • 计数器（C） • 程序块（FC/FB/SFC/SFB） • 数据块（DB） • 用户定义数据类型（UDT） • 变量表（VAT）	可以为下列对象定义局域变量： • 块参数 • 块的静态数据 • 块的临时数据
定义位置	符号表	程序块的变量声明区

【例 4-2】　将如图 4-3 所示的绝对寻址的起停控制梯形图，换成符号寻址梯形图。



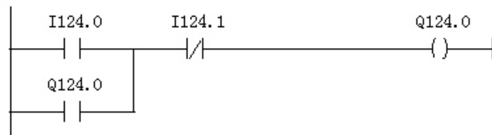


图 4-3　绝对寻址的梯形图

【解】

打开 STEP 7 的项目管理器，先选中“S7 程序（1）”，再双击“符号”，如图 4-4 所示，弹出符号编辑器界面，输入如图 4-5 所示的信息，最后单击工具栏的“保存”按钮，将输入的符号分配给相应的地址，例如“起动”分配给地址“I124.0”。再打开程序编辑器，符号寻址的梯形图如图 4-6 所示。

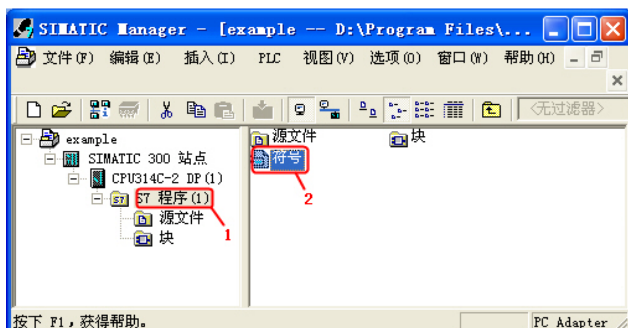


图 4-4　项目管理器界面



	状态	符号	地址		数据类型	注释
1		启动	I	124.0	BOOL	
2		停止	I	124.1	BOOL	
3		电动机	Q	124.0	BOOL	
4						

图 4-5　符号编辑器界面

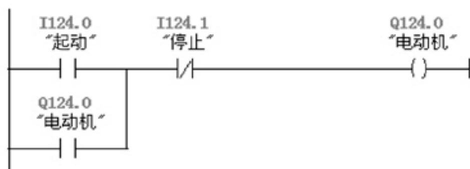


图 4-6　符号寻址的梯形图

如图 4-7 所示，功能块的 IN（输入引脚）上的“Sw_On”和“Sw_Off”数据类型为

Bool，是局域变量，其有效范围仅在 FB1 功能块中。

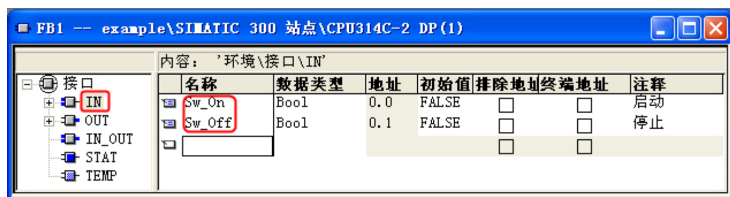


图 4-7　局域变量

3．间接寻址

（1）存储器间接寻址

在存储器间接寻址指令中，给出一个地址指针的存储器，该存储器的内容是操作数所在存储单元的地址。在循环程序中经常用到存储器间接寻址。

地址指针可以是字或双字，定时器（T）、计数器（C）、数据块（DB）、功能块（FB）和功能（FC）的编号范围小于 65535，使用字指针就可以。其他地址则要使用双字指针，如果要用双字格式的指针访问一个字、字节或双字存储器，必须保证指针的位编号为 0，例如 P#Q20.0。

存储器间接寻址的双字指针格式如图 4-8 所示，其中 0～2 位为被寻址地址中的位编号，3～18 位为寻址字节编号。只有 M、L、DB、PI 存储区域的双字节才能做地址指针。

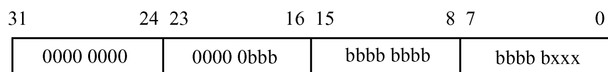


图 4-8　存储器间接寻址的双字指针格式

存储器间接寻址应用如下：

L QB[DBD 10] //将输出字节装入累加器 1，输出字节的地址指针在数据双字 DBD10 中，如果 DBD10 的值为 2＃0000 0000 0000 0000 0000 0000 0010 0000，装入的是 QB4

A M[LD 4] //对存储器位作“与”运算，地址指针在数据双字 LD4 中，如果 LD4 的值为 2＃0000 0000 0000 0000 0000 0000 0010 0011，则是对 M4.3 进行操作

（2）寄存器间接寻址

地址寄存器 AR1 和 AR2，它们中的内容加上偏移量形成地址指针，指向数值所在的存储单元。寄存器间接寻址中双字指针格式如图 4-9 所示。

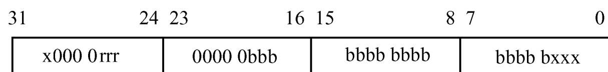


图 4-9　寄存器间接寻址的双字指针格式

其中第 0～2 位（xxx）为被寻址地址中位的编号（0～7），第 3～18 位为被寻址地址的字节的编号（0～65535），第 24～26 位（rrr）为被寻址地址的区域标识号，第 31 位 x = 0 为区域内的间接寻址，第 31 位 x = 1 为区域间的间接寻址。寄存器间接寻址的区域标识位见表 4-5。



表 4-5　寄存器间接寻址的区域标识位

区域标识符	存储区	位 26～24
P	外设输入/输出	000
I	输入过程映像	001
Q	输出过程映像	010
M	位存储区	011
DBX	共享数据块	100
DIX	背景数据块	101
L	块的局域数据	111

STEP 7 中有两种格式的寄存器间接寻址方式，分别是区域内的间接寻址和区域间的间接寻址。当 31 位为 0 时，为区域内的间接寻址；当 31 位为 1 时，为区域间的间接寻址。

第一种地址指针格式存储区的类型在指令中给出，例如 LDBB[AR1, P#6.0]。在某一存储区内寻址。第 24～26 位（rrr）应为 0。

第二种地址指针格式的第 24～26 位还包含存储区域标识符 rrr，区域间寄存器间接寻址。

如果要用寄存器指针访问一个字节、字或双字，必须保证指针中的位地址编号为 0。

指针常数＃P5.0 对应的二进制数为 2＃0000 0000 0000 0000 0000 0000 0010 1000。

下面是区内间接寻址的例子：

```
L P#5.0                 //将间接寻址的指针装入累加器 1
LAR1                    //将累加器 1 中的内容送到地址寄存器 1
A M[AR1, P#2.3]         //AR1 中的 P#5.0 加偏移量 P#2.3，实际上是对 M7.3 进行操作
= Q[AR1, P#0.2]         //逻辑运算的结果送 Q5.2
L DBW[AR1, P#18.0]      //将 DBW23 装入累加器 1
```

下面是区域间间接寻址的例子：

```
L P#M6.0                //将存储器位 M6.0 的双字指针装入累加器 1
LAR1                    //将累加器 1 中的内容送到地址寄存器 1
T W[AR1, P#50.0]        //将累加器 1 的内容传送到存储器字 MW56
```

P#M6.0 对应的二进制数为 2#1000 0011 0000 0000 0000 0000 0011 0000。因为地址指针 P#M6.0 中已经包含有区域信息，使用间接寻址的指令 T W[AR1, P#50]中没有必要再用地址标识符 M。

4.1.3　编程语言

1．PLC 编程语言的国际标准

IEC 61131 是 PLC 的国际标准，1992～1995 年发布了 IEC 61131 标准中的 1～4 部分，我国在 1995 年 11 月发布了 GB/T15969-1/2/3/4（等同于 IEC 61131-1/2/3/4）。

IEC 61131-3 广泛地应用于 PLC、DCS 和工控机、“软件 PLC”、数控系统、RTU 等产品。其定义了 5 种编程语言，分别是指令表（Instruction list，IL）、结构文本（Structured





text，ST）、梯形图（Ladder diagram，LD）、功能块图（Function block diagram，FBD 和 顺序功能图（Sequential function chart，SFC）。

2．STEP 7 中的编程语言

STEP 7 中有梯形图、语句表和功能块图 3 种基本编程语言，可以相互转换。通过安装软件包，还有其他的编程语言，以下简要介绍。

（1）顺序功能图（SFC）

STEP 7 中为 S7 Graph，它不是 STEP 7 的标准配置，需要安装软件包，S7 Graph 是针对顺序控制系统进行编程的图形编程语言，特别适合顺序控制程序编写。

（2）梯形图（LAD）

梯形图直观易懂，适合于数字量逻辑控制。“能流”（Power flow）与程序执行的方向。梯形图适合于熟悉继电器电路的人员使用。设计复杂的触点电路时最好用梯形图。其应用最为广泛。

（3）语句表（STL）

语句表功能比梯形图或功能块图的功能强。语句表可供喜欢用汇编语言编程的用户使用。语句表输入快，可以在每条语句后面加上注释。设计高级应用程序时建议使用语句表。

（4）功能块图（FBD）

“LOGO！”系列微型 PLC 使用功能块图编程。功能块图适合于熟悉数字电路的人员使用。

（5）结构文本（ST）

STEP 7 的 S7 SCL（结构化控制语言）符合 EN61131-3 标准。SCL 适合于复杂的公式计算、复杂的计算任务和最优化算法或管理大量的数据等。S7 SCL 编程语言适合于熟悉高级编程语言（例如 PASCAL 或 C 语言）的人员使用。它不是 STEP 7 的标准配置，需要安装软件包。

（6）S7 HiGraph 编程语言

图形编程语言 S7 HiGraph 属于可选软件包，它用状态图（Stategraphs）来描述异步、非顺序过程的编程语言。HiGraph 适合于异步非顺序过程的编程。

（7）S7 CFC 编程语言

可选软件包 CFC（Continuous Function Chart，连续功能图）用图形方式连接程序库中以块的形式提供的各种功能。CFC 适合于连续过程控制的编程。它不是 STEP 7 的标准配置，需要安装软件包。

在 STEP 7 编程软件中，如果程序块没有错误，并且被正确地划分为网络，在梯形图、功能块图和语句表之间可以转换。如果部分网络不能转换，则用语句表表示。

4.2　CPU中的寄存器

4.2.1　累加器（ACCUx）

累加器是用于处理字节、字或双字的寄存器。S7-300 PLC 有两个 32 位累加器（ACCU1



和 ACCU2），S7-400 PLC 有 4 个累加器（ACCU1～ACCU4）。数据放在累加器的低端（右对齐）。

4.2.2 状态字寄存器（16 位）

状态字的结构如图 4-10 所示，以下将详述各位的含义。

15~9	8	7	6	5	4	3	2	1	0
未用	BR	CC1	CC0	OS	OV	OR	STA	RIO	$\overline{FC}$

图 4-10 状态字的结构

首次检测位/FC，逻辑运算结果（RLO）。

状态位 STA 不能用指令检测。

OR 位暂存逻辑“与”的操作结果（先“与”后“或”）。

算术运算或比较指令执行时出现错误，溢出位 OV 被置 1。

OV 位被置 1 时，溢出状态保持为 1，OS 位也被置 1；OV 位被清 0 时，OS 仍保持为 1，用于指明前面的指令执行过程中是否产生过错误。

条件码 1（CC1）和条件码 0（CC0）综合起来用于表示在累加器 1 中产生的算术运算或逻辑运算的结果与 0 的大小关系、比较指令的执行结果或移位指令的移出位状态。

二进制结果位（BR）在一段既有位操作又有字操作的程序中，用于表示字操作结果是否正确。在梯形图的方框指令中，BR 位与 ENO 有对应关系，用于表明方框指令是否被正确执行。如果执行出现了错误，BR 位为 0，ENO 也为 0；如果功能被正确执行，BR 位为 1，ENO 也为 1。

4.2.3 数据块寄存器

DB 和 DI 寄存器分别用来保存打开的共享数据块和背景数据块的编号。

4.3 位逻辑指令

位逻辑指令用于二进制数的逻辑运算。位逻辑运算的结果简称为 RLO。

位逻辑指令是最常用的指令之一，主要有与指令、与非指令、或指令、或非指令、置位指令、复位指令和输出指令等。

1．触点与线圈

A（And）：与指令表示串联的常开触点，检测信号 1，与 And 关联。

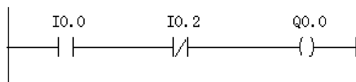
O（Or）：或指令表示并联的常开触点，检测信号 1，与 Or 关联。

AN（And Not）：与非指令表示串联的常闭触点，检测信号 0，与 And Not 关联。

ON（Or Not）：或非指令表示并联的常闭触点，检测信号 0，与 Or Not 关联。

输出指令“=”将操作结果 RLO 赋值给地址位，与线圈相对应。

与、与非及输出指令示例如图 4-11 所示，图中左侧是梯形图，右侧是与梯形图对应的指令表。当常开触点 I0.0 和常闭触点 I0.2 都接通时，输出线圈 Q0.0 得电（Q0.0＝1），Q0.0＝1 实际上就是运算结果 RLO 的数值，I0.0 和 I0.2 是串联关系。

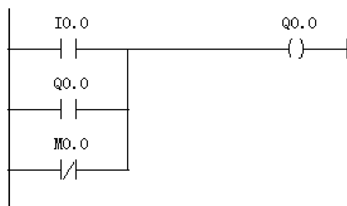


程序段 1：与、与非

A	I	0.0
AN	I	0.2
=	Q	0.0

图 4-11　与、与非及输出指令示例

或、或非及输出指令示例如图 4-12 所示，当常开触点 I0.0、常开触点 Q0.0 和常闭触点 M0.0 有一个接通时，输出线圈 Q0.0 得电（Q0.0＝1），I0.0、Q0.0 和 M0.0 是并联关系。



程序段1：或、或非

O	I	0.0
O	Q	0.0
ON	M	0.0
=	Q	0.0

图 4-12　或、或非及输出指令示例

【例 4-3】 设计电动机的“正转－停－反转”的梯形图，其中 I0.0 是正转按钮、I0.1 是反转按钮、I0.2 是停止按钮、Q0.0 是正转输出、Q0.1 是反转输出。

【解】

先设计 PLC 的 I/O 接线图，如图 4-13 所示。

借鉴继电器接触器系统中的设计方法，不难设计“正转－停－反转”梯形图，如图 4-14 所示。常开触点 Q0.0 和常开触点 Q0.1 起自保（自锁）作用，而常闭触点 Q0.0 和常闭触点 Q0.1 起互锁作用。

梯形图中虽然有 Q0.0 和 Q0.1 常闭触点互锁，但由于 PLC 的扫描速度极快，Q0.0 的断开和 Q0.1 的接通几乎是同时发生的，若 PLC 的外围电路无互锁触点，就会使正转接触器断开，其触点间电弧未灭时，反转接触器已经接通，可能导致电源瞬时短路。为了避免这种情况的发生，外部电路需要互锁，图 4-13 用 KM1 和 KM2 实现这一功能。正反转切换时，最好能延时一段时间。想一想，若停止按钮与常开触点相连，则梯形图应该做何变化？

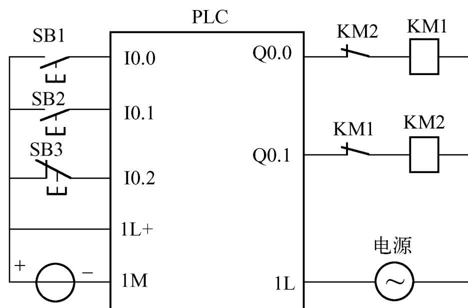
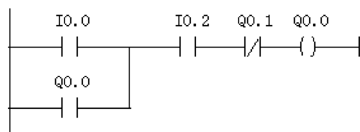


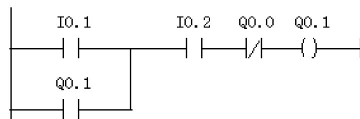
图 4-13　I/O 接线图



程序段 1：正转



程序段 2：反转




```
程序段1：正转
A(
  O    I    0.0
  O    Q    0.0
  )
  AN   I    0.2
  AN   Q    0.1
  =    Q    0.0
程序段2：反转
A(
  O    I    0.1
  O    Q    0.1
  )
  AN   I    0.2
  AN   Q    0.0
  =    Q    0.1
```

图 4-14 “正转—停—反转”梯形图

2．对RLO的直接操作指令

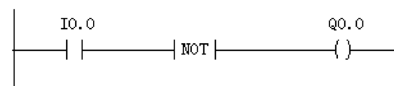
这类指令可直接对逻辑操作结果 RLO 进行操作，改变状态字中 RLO 的状态。对 RLO 的直接操作指令见表 4-6。

表 4-6　对 RLO 的直接操作指令

梯形图指令	STL 指令	功能说明	说明
---|NOT|---	NOT	取反 RLO	在逻辑串中，对当前 RLO 取反
	SET	置位 RLO	将 RLO 置 1
	CLR	复位 RLO	将 RLO 清零
—（SAVE）	SAVE	保存 RLO	将 RLO 保存到状态字的 BR 位

取反触点示例如图 4-15 所示，当 I0.0 为 1 时 Q0.0 为 0，反之当 I0.0 为 0 时 Q0.0 为 1。

程序段 1：取反触点



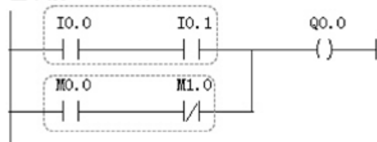

```
程序段1：取反触点
A      I      0.0
NOT
=      Q      0.0
```

图 4-15　取反触点示例

3．电路块的串联和并联

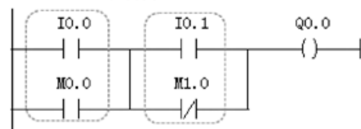
与 S7-200 PLC 不同，S7-300/400 PLC 的电路块没有专用的指令。如图 4-16 所示的并联块，实际就是把两个虚线框当做两个块，再将两个块做或运算。如图 4-17 所示的串联块，实际就是把两个虚线框当做两个块，再将两个块做与运算。

程序段 1：并联块




```
A    I    0.0
A    I    0.1
O
A    M    0.0
AN   M    1.0
=    Q    0.0
```

图 4-16　并联块示例




```
A(
O   I   0.0
O   M   0.0
)
A(
O   I   0.1
ON  M   1.0
)
=   Q   0.0
```

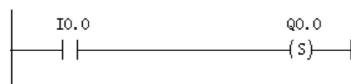
图 4-17 串联块示例

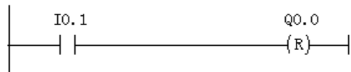
4．复位与置位指令

S：置位指令将指定的地址位置位（变为 1，并保持）。

R：复位指令将指定的地址位复位（变为 0，并保持）。

如图 4-18 所示为置位/复位指令应用例子，当 I0.0 为 1，Q0.0 为 1，之后，即使 I0.0 为 0，Q0.0 保持为 1，直到 I0.1 为 1 时，Q0.0 变为 0。这两条指令非常有用。





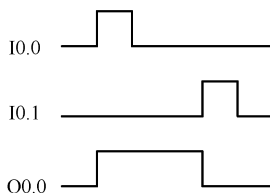


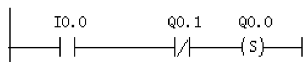
图 4-18 置位/复位指令示例

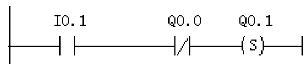
【关键点】置位/复位指令不一定要成对使用。

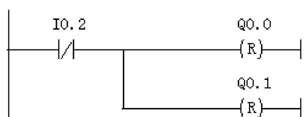
【例 4-4】 用置位/复位指令编写“正转－停－反转”的梯形图，其中 I0.0 是正转按钮、I0.1 是反转按钮、I0.2 是停止按钮、Q0.0 是正转输出、Q0.1 是反转输出。

【解】

梯形图和指令表如图 4-19 所示，可见使用置位/复位指令后，不需要用自锁，程序变得更加简洁。








```
程序段1：正转
  A   I   0.0
  AN  Q   0.1
  S   Q   0.0
程序段2：反转
  A   I   0.1
  AN  Q   0.0
  S   Q   0.1
程序段3：停止
  AN  I   0.2
  R   Q   0.0
  R   Q   0.1
```

图 4-19 “正转－停－反转”梯形图



5．RS /SR双稳态触发器

RS：置位优先型 RS 双稳态触发器。如果 R 输入端的信号状态为“1”，S 输入端的信号状态为“0”，则复位 RS（置位优先型 RS 双稳态触发器）。否则，如果 R 输入端的信号状态为“0”，S 输入端的信号状态为“1”，则置位触发器。如果两个输入端的 RLO 状态均为“1”，则指令的执行顺序是最重要的。RS 触发器先在指定地址执行复位指令，然后执行置位指令，以使该地址在执行余下的程序扫描过程中保持置位状态。RS /SR 双稳态触发器示例如图 4-20 所示，用一个表格表示这个例子的输入与输出的对应关系，见表 4-7。

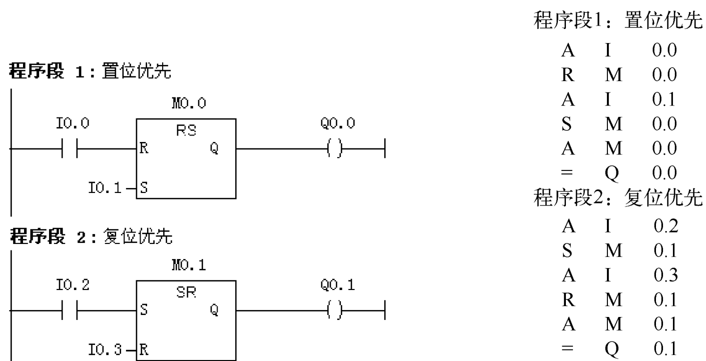


图 4-20　RS /SR 双稳态触发器示例

图 4-7　RS /SR 双稳态触发器输入与输出的对应关系

置位优先 RS				复位优先 SR			
输入状态		输出状态	说明	输入状态		输出状态	说明
I0.0	I0.1	Q0.0	当各个状态断开后，输出状态保持	I0.2	I0.3	Q0.1	当各个状态断开后，输出状态保持
1	0	0		1	0	1	
0	1	1		0	1	0	
1	1	1		1	1	0	

SR：复位优先型 SR 双稳态触发器。如果 S 输入端的信号状态为“1”，R 输入端的信号状态为“0”，则置位 SR （复位优先型 SR 双稳态触发器）。否则，如果 S 输入端的信号状态为“0”，R 输入端的信号状态为“1”，则复位触发器。如果两个输入端的 RLO 状态均为“1”，则指令的执行顺序是最重要的。SR 触发器先在指定地址执行置位指令，然后执行复位指令，以使该地址在执行余下的程序扫描过程中保持复位状态。

【例 4-5】 用 SR 双稳态触发器指令编写“正转－停－反转”的梯形图，其中 I0.0 是正转按钮、I0.1 是反转按钮、I0.2 是停止按钮、Q0.0 是正转输出、Q0.1 是反转输出。

【解】

先设计其接线图如图 4-21 所示。

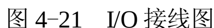


图 4-21　I/O 接线图

梯形图和指令表如图 4-22 所示，可见使用 SR 双稳态触发器指令后，不需要用自锁，程序变得更加简洁。当按下按钮 I0.2 后，由于复位优先，电动机无论正转或者反转都会停下，当复位按钮未按下，且电动机处于停止状态时，按下 I0.0 按钮电动机正转，按下 I0.1 按钮电动机反转。

```
程序段1：正转、停
A    I    0.0
AN   Q    0.1
S    M    0.0
A    I    0.2
R    M    0.0
A    M    0.0
=    Q    0.0
程序段2：正转、停
A    I    0.1
AN   Q    0.0
S    M    0.1
A    I    0.2
R    M    0.1
A    M    0.1
=    Q    0.1
```

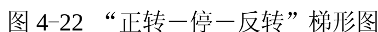


图 4-22 “正转－停－反转”梯形图

6．边沿检测指令

边沿检测指令有负跳沿检测指令（下降沿检测）和正跳沿检测（上升沿检测）指令。

负跳沿检测指令 FN 检测 RLO 从 1 调转到 0 时的下降沿，并保持 RLO＝1 一个扫描周期。每个扫描周期期间，都会将 RLO 位的信号状态与上一个周期获取的状态比较，以判断是否改变。

```
程序段1：下降沿
A    I    0.0
FN   M    0.0
=    Q    0.0
```

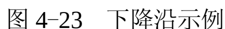


图 4-23　下降沿示例

下降沿示例的梯形图和指令表如图 4-23 所示，由如图 4-24 所示的时序图可知：当按钮 I0.0 按下后弹起时，产生一个下降沿，输出 Q0.0 得电一个扫描周期，这个时间是很短的，肉眼是分辨不出来的，因此若 Q0.0 控制的是一盏灯，肉眼是不能分辨出灯已经亮了一个扫



描周期。在后面的章节中多处用到时序图，请读者务必学会这种表达方式。

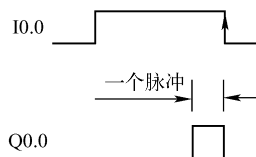


图 4-24　下降沿示例时序图

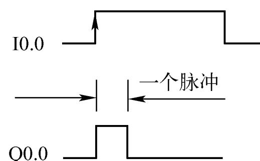
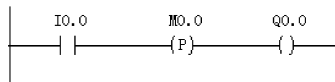


图 4-25　上升沿示例时序图

正跳沿检测指令 FP 检测 RLO 从 0 调转到 1 时的上升沿，并保持 RLO＝1 一个扫描周期。每个扫描周期期间，都会将 RLO 位的信号状态与上一个周期获取的状态比较，以判断是否改变。

上升沿示例的梯形图和指令表如图 4-26 所示，由如图 4-25 所示的时序图可知：当按钮 I0.0 按下时，产生一个上升沿，输出 Q0.0 得电一个扫描周期，无论按钮闭合多漫长的时间，输出 Q0.0 只得电一个扫描周期。



程序段1：上升沿

```
A    I    0.0
FP   M    0.0
=    Q    0.0
```

图 4-26　上升沿示例

【例 4-6】 边沿检测指令应用梯形图如图 4-27 所示，如果按钮 I0.0 压下闭合 1s 后弹起，请分析程序运行结果。

【解】

时序图如图 4-28 所示，当 I0.0 压下时，产生上升沿，触点产生一个扫描周期的时钟脉冲，驱动输出线圈 Q0.1 通电一个扫描周期，Q0.0 也通电，使输出线圈 Q0.0 置位，并保持。

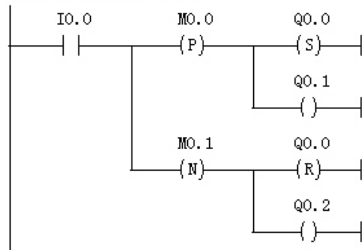


图 4-27　边沿检测指令示例

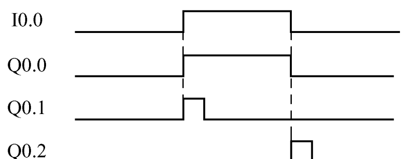


图 4-28　边沿检测指令示例时序图

当按钮 I0.0 弹起时，产生下降沿，触点产生一个扫描周期的时钟脉冲，驱动输出线圈 Q0.2 通电一个扫描周期，使输出线圈 Q0.0 复位，并保持，Q0.0 得电共 1s。

4.4　定时器与计数器指令

4.4.1　定时器

STEP 7 的定时器指令相当于继电器接触器控制系统的时间继电器的功能。定时器的数量随 CPU 的类型不同，从 32 个到 512 个不等，一般而言足够用户使用。

1．定时器的种类

STEP 7 的定时器指令较为丰富，除了常用的接通延时定时器（SD）和断开延时定时器（SF）以外，还有脉冲定时器（SP）、扩展脉冲定时器（SE）和保持型接通延时定时器（SS）共 5 类。

2．定时器的使用

定时器有其存储区域，每个定时器有一个 16 位的字和一个二进制的值。定时器的字存放当前定时值。二进制的值表示定时器的接点状态。

（1）启动和停止定时器

在梯形图中，定时器的 S 端子可以使能定时器，而定时器的 R 端子可以复位定时器。

（2）设定定时器的定时时间

STEP 7 中的定时时间由时基和定时值组成，定时时间为时基和定时值的乘积，例如定时值为 1000，时基为 0.01s，那么定时时间就是 10s，很多 PLC 的定时都是采用这种方式。定时器开始工作后 ，定时值不断递减，递减至零，表示时间到，定时器会相应动作。

定时器字的格式如图 4−29 所示，其中第 12 和 13 位（即 m 和 n）是定时器的时基代码，时基代码的含义见表 4−8。定时的时间值以 3 位 BCD 码格式存放，位于 0～11（即 a~l），范围为 0～999。第 14 位和 15 位不用。

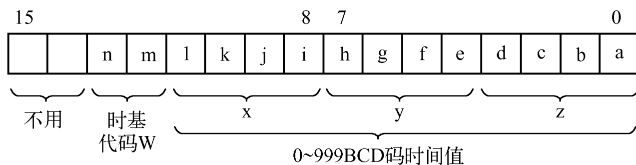


图 4−29　定时器字的格式

定时时间有两种表达方式，十六进制数表示和 S5 时间格式表示。前者的格式为：W#16#wxyz，其中 w 是时间基准代码，xyz 是 BCD 码的时间值。例如时间表述为：W#16#1222，则定时时间为 222×0.1s=22.2s。

表 4−8　时基与定时范围对应表

时基二进制代码	时基	分辨率（S）	定时范围
00	10ms	0.01	10ms~9s_990ms
01	100ms	0.1	100ms~1m_39s_900ms
10	1s	1	1s~16m_39s
11	10s	10	10s~2h_46m_30s



S5 时间格式为：S5T#aH_bM_cS_dMS，其中 a 表示小时，b 表示分钟，c 表示秒钟，d 表示毫秒，含义比较明显。例如 S5T#1H_2M_3S 表示定时时间为 1 小时 2 分 3 秒。这里的时基是 PLC 自动选定的。

3．脉冲时间定时器（SP）

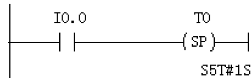
SP：产生指定时间宽度脉冲的定时器。当逻辑位有上升沿时，脉冲定时器指令启动计时，同时节点立即输出高电平“1”，直到定时器时间到，定时器输出为“0”。脉冲时间定时器可以将长信号变成指定宽度的脉冲。如果定时时间未到，而逻辑位的状态变成“0”时，定时器停止计时，输出也变成低电平。脉冲的定时器线圈指令和参数见表 4-9。

表 4-9　脉冲定时器线圈指令和参数

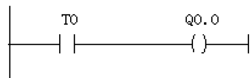
LAD	参数	数据类型	存储区	说明
T no. —（SP）	T no.	TIMER	T	表示要启动的定时器号
	时间值	S5TIME	I、Q、M、D、L	定时器时间值

用一个例子说明脉冲定时器的使用，梯形图如图 4-30 所示，对应的时序图如图 4-31 所示，　可以看出当 I0.0 接通的时间长于 1s，Q0.0 输出 1 的时间是 1s，而当 I0.0 接通的时间为 0.5s（小于 1s）时，Q0.0 输出 1 的时间是 0.5s，无论 I0.0 是否接通，只要 I0.1 接通时，定时器复位，Q0.0 输出为 0。

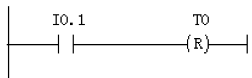
程序段 1：标题：



程序段 2：标题：



程序段 3：标题：




```
程序段1：标题：
    A    I   0.0
    L    S5 T#1S
    SP   T  0
程序段2：标题：
    A    T    0
    =    Q    0.0
程序段3：标题：
    A    I    0.1
    R    T    0
```

图 4-30　脉冲定时器示例

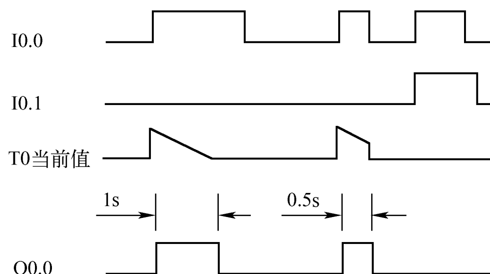


图 4-31　脉冲定时器示例的时序图





STEP 7 除了提供脉冲的定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。脉冲定时器方框指令和参数见表 4-10。

表 4-10　脉冲定时器方框指令和参数

LAD	参数	数据类型	说明	存储区
T no. S_PULSE S　Q TV　BI R　BCD	T no.	TIMER	要启动的计时器号，如 T0	T
	S	BOOL	启动输入端	I，Q，M，D，L
	TV	S5TIME	定时时间（S5TIME 格式）	
	R	BOOL	复位输入端	
	Q	BOOL	定时器的状态	
	BI	WORD	当前时间（整数格式）	
	BCD	WORD	当前时间（BCD 码格式）	

脉冲定时器方框指令的示例如图 4-32 所示，本例时序图和运行结果与图 4-31 所示是完全相同的，而且程序更为简洁。

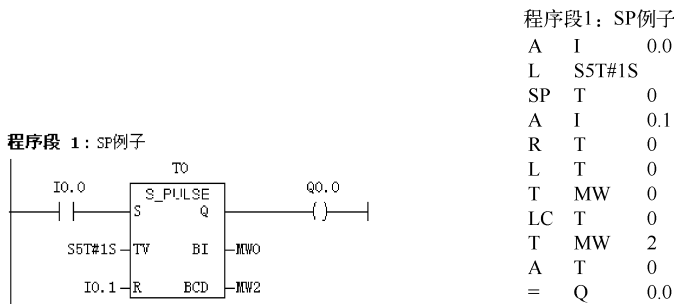


图 4-32　脉冲定时器方框指令示例

4．扩展脉冲时间定时器（SE）

扩展脉冲时间定时器（SE）和脉冲时间定时器（SP）指令相似，但 SE 指令具有保持功能。扩展脉冲时间定时器的线圈指令和参数见表 4-11。

表 4-11　扩展脉冲定时器线圈指令和参数

LAD	参数	数据类型	存储区	说明
T no. —（SE）	T no.	TIMER	T	表示要启动的定时器号
	时间值	S5TIME	I、Q、M、D、L	定时器时间值

用一个例子来说明 SE 线圈指令的使用，梯形图和指令表如图 4-33 所示，对应的时序图如图 4-34 所示。当 I0.0 有上升沿时，定时器 T0 启动，同时 Q0.0 输出高电平“1”，定时时间到后，输出自动变为“0”（尽管此时 I0.0 仍然闭合），当 I0.0 有上升沿时，且闭合时间没有到定时时间，Q0.0 仍然输出为“1”，直到定时时间到为止。无论什么情况下，只要复



位输入端起作用，本例为 I0.1 闭合，则定时器复位，输出为“0”。

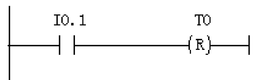
程序段 1：标题:



程序段 2：标题:



程序段 3：标题:




```
程序段1：标题：
A    I    0.0
L    S5T#1S
SE   T    0
程序段2：标题：
A    T    0
=    Q    0.0
程序段3：标题：
A    I    0.1
R    T    0
```

图 4-33　扩展脉冲定时器示例

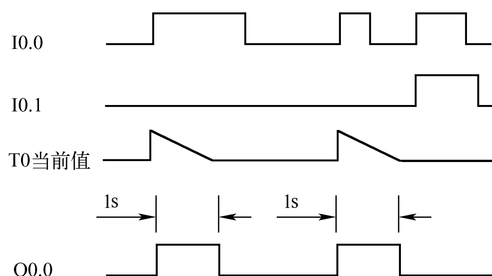


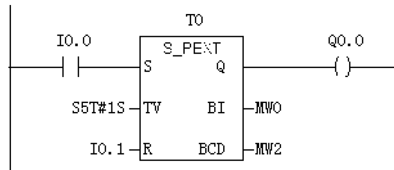
图 4-34　扩展脉冲定时器示例的时序图

STEP 7 除了提供扩展脉冲的定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。扩展脉冲定时器方框指令和参数见表 4-12。

表 4-12　扩展脉冲定时器方框指令和参数

LAD	参数	数据类型	说明	存储区
T no. S_PEXT S　Q TV　BI R　BCD	T no.	TIMER	要启动的定时器号，如 T0	T
	S	BOOL	启动输入端	I，Q，M，D，L
	TV	S5TIME	定时时间（S5TIME 格式）	
	R	BOOL	复位输入端	
	Q	BOOL	定时器的状态	
	BI	WORD	当前时间（整数格式）	
	BCD	WORD	当前时间（BCD 码格式）	

扩展脉冲定时器方框指令的示例如图 4-35 所示，本例时序图和运行结果与图 4-32 所示的是完全相同的，而且程序更为简洁。



程序段1：SE例子

```
A   I    0.0
L   S5T#1S
SE  T    0
A   I    0.1
R   T    0
L   T    0
T   MW   0
LC  T    0
T   MW   2
A   T    0
=   Q    0.0
```

图 4-35　扩展脉冲定时器方框指令示例

5．接通延时定时器（SD）

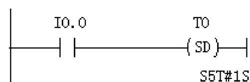
接通延时定时器（SD）相当于继电器接触器控制系统中的通电延时时间继电器。通电延时继电器的工作原理是：线圈通电，触点延时一段时间后动作。SD 指令是当逻辑位接通时，定时器开始定时，计时过程中，定时器的输出为“0”，定时时间到，输出为“1”，整个过程中，逻辑位要接通，只要逻辑位断开，则输出为“0”。接通延时定时器最为常用。接通延时定时器的线圈指令和参数见表 4-13。

表 4-13　接通延时定时器线圈指令和参数

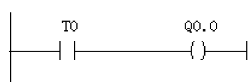
LAD	参数	数据类型	存储区	说明
T no. —（SD）	T no.	TIMER	T	表示要启动的定时器号
	时间值	S5TIME	I、Q、M、D、L	定时器时间值

用一个例子来说明 SD 线圈指令的使用，梯形图和指令表如图 4-36 所示，对应的时序图如图 4-37 所示。当 I0.0 闭合时，定时器 T0 开始定时，定时 1s 后（I0.0 一直闭合），Q0.0 输出高电平“1”，若 I0.0 的闭合时间不足 1s，Q0.0 输出为“0”，若 I0.0 断开，Q0.0 输出为“0”。无论什么情况下，只要复位输入端起作用，本例为 I0.1 闭合，则定时器复位，Q0.0 输出为“0”。

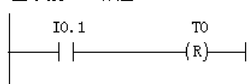
程序段 1：标题：



程序段 2：标题：



程序段 3：标题：




```
程序段1
A   I    0.0
L   S5T#1S
SD  T    0
程序段2
A   T    0
=   Q    0.0
程序段3
I   I   0.1
T   T   0
```

图 4-36　接通延时定时器示例



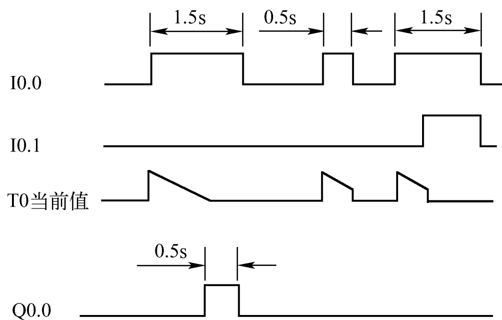


图 4-37　接通延时定时器示例的时序图

STEP 7 除了提供接通延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。接通延时定时器方框指令和参数见表 4-14。

表 4-14　接通延时定时器方框指令和参数

LAD	参数	数据类型	说明	存储区
T no. S_ODT S　Q TV　BI R　BCD	T no.	TIMER	要启动的定时器号，如 T0	T
	S	BOOL	启动输入端	I，Q，M，D，L
	TV	S5TIME	定时时间（S5TIME 格式）	
	R	BOOL	复位输入端	
	Q	BOOL	定时器的状态	
	BI	WORD	当前时间（整数格式）	
	BCD	WORD	当前时间（BCD 码格式）	

接通延时定时器方框指令的示例如图 4-38 所示，本例时序图和运行结果和图 4-35 所示的是完全相同的，而且程序更为简洁。

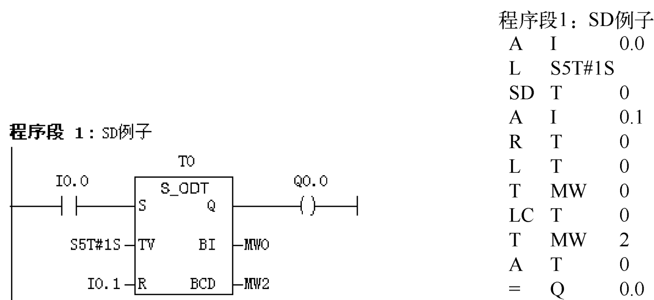


图 4-38　接通延时定时器方框指令示例

【例 4-7】 用 S7-300 PLC 控制一盏灯的闪烁，闪烁频率为 1Hz，要求用接通延时定时器，请设计梯形图。

【解】

梯形图如图 4-39 所示。这个梯形图比较简单，但初学者往往不易看懂。控制过程是：当 I0.0 合上定时器 T0 定时 0.5s 后 Q0.0 控制的灯亮，与此同时定时器 T1 启动定时，0.5s

后，T1 的常闭触点断开切断 T0，进而 T0 的常开触点切断 T1 和 Q0.0，灯灭；此时 T1 的常闭触点闭合 T0 又开始定时，如此周而复始，Q0.0 控制灯闪烁。这个问题还有其他的解决方案，由读者完成。

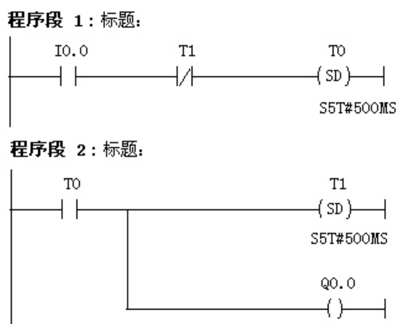


图 4-39　梯形图

6．保持型接通延时定时器（SS）

保持型接通延时定时器（SS）与接通延时定时器（SD）类似，但 SS 定时器具有保持功能。一旦逻辑位有上升沿发生，定时器启动计时，延时时间到，输出高电平“1”，即使逻辑位为“0”也不影响定时器的工作。必须用复位指令才能使定时器复位。保持型接通延时定时器的线圈指令和参数见表 4-15。

表 4-15　保持型接通延时定时器线圈指令和参数

LAD	参数	数据类型	存储区	说明
T no. —（SS）	T no.	TIMER	T	表示要启动的定时器号
	时间值	S5TIME	I、Q、M、D、L	定时器时间值

用一个例子来说明 SS 线圈指令的使用，梯形图和指令表如图 4-40 所示，对应的时序图如图 4-41 所示。当 I0.0 闭合产生一个上升沿时，定时器 T0 开始定时，定时 1s 后（无论 I0.0 是否闭合），Q0.0 输出为高电平“1”，直到复位有效为止，本例为 I0.1 闭合产生上升沿，定时器复位，Q0.0 输出为低电平“0”。

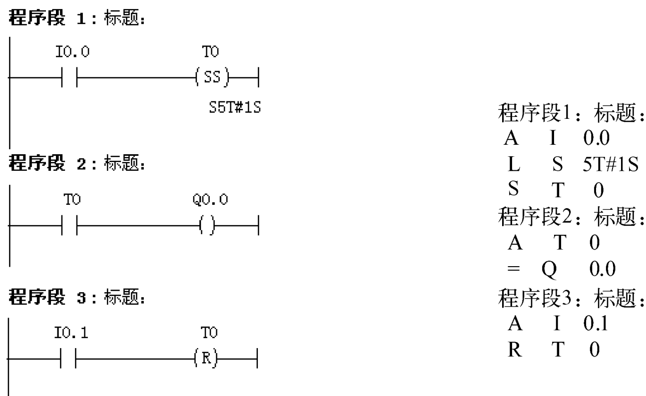


图 4-40　保持型接通延时定时器示例



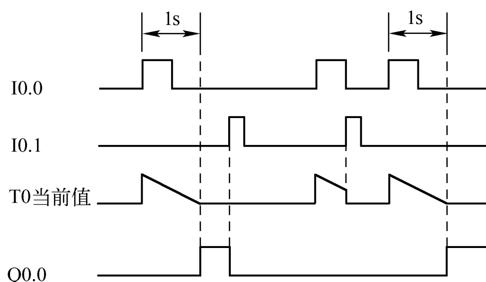


图 4-41　保持型接通延时定时器示例的时序图

STEP 7 除了提供保持型接通延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。保持型接通延时定时器方框指令和参数见表 4-16。

表 4-16　保持型接通延时定时器方框指令和参数

LAD	参数	数据类型	说明	存储区
T no. S_ODTS S　Q TV　BI R　BCD	T no.	TIMER	要启动的定时器号，如 T0	T
	S	BOOL	启动输入端	I，Q，M，D，L
	TV	S5TIME	定时时间（S5TIME 格式）	
	R	BOOL	复位输入端	
	Q	BOOL	定时器的状态	
	BI	WORD	当前时间（整数格式）	
	BCD	WORD	当前时间（BCD 码格式）	

保持型接通延时定时器方框指令的示例如图 4-42 所示，本例时序图和运行结果与图 4-39 所示的是完全相同的，而且程序更为简洁。

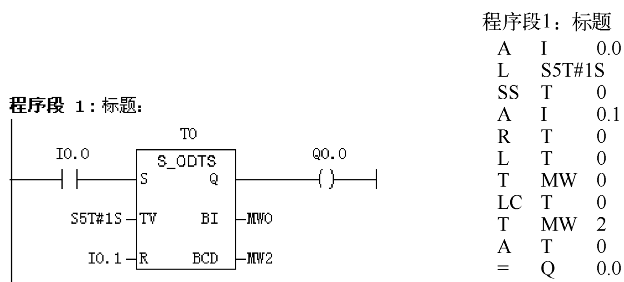


图 4-42　保持型接通延时定时器方框指令的示例

7．断开延时定时器（SF）

断开延时定时器（SF）相当于继电器控制系统的断电延时时间继电器，是定时器指令中唯一一个由下降沿启动的定时器指令。断开延时定时器的线圈指令和参数见表 4-17。

表 4-17　断开延时定时器线圈指令和参数

LAD	参数	数据类型	存储区	说明
T no. —（SF）	T no.	TIMER	T	表示要启动的定时器号
	时间值	S5TIME	I、Q、M、D、L	定时器时间值

用一个例子来说明 SF 线圈指令的使用，梯形图和指令表如图 4-43 所示，对应的时序图如图 4-44 所示。当 I0.0 闭合时，Q0.0 输出高电平“1”，当 I0.0 断开时产生一个下降沿，定时器 T0 开始定时，定时 1s 后（无论 I0.0 是否闭合），定时时间到，Q0.0 输出为低电平“0”。任何时候复位有效时，定时器 T0 定时停止，Q0.0 输出为低电平“0”。

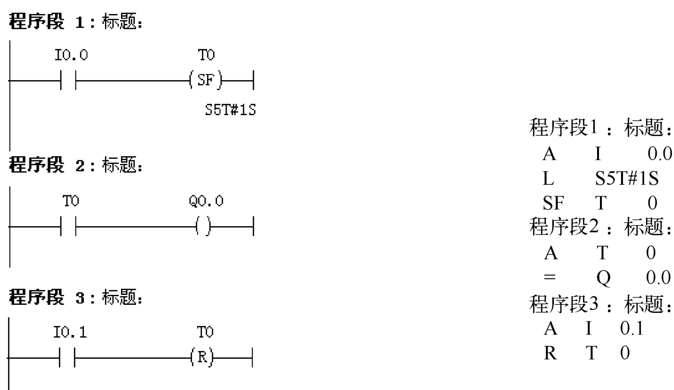


图 4-43　断开延时定时器示例

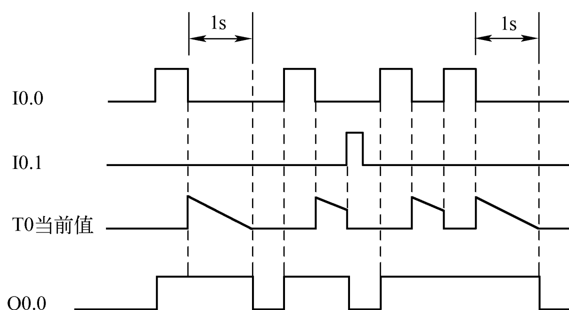


图 4-44　断开延时定时器示例的时序图

STEP 7 除了提供断开延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。断开延时定时器方框指令和参数见表 4-18。

表 4-18　断开延时定时器方框指令和参数

LAD	参数	数据类型	说明	存储区
T no. S_OFFDT S　Q TV　BI R　BCD	T no.	TIMER	要启动的定时器号，如 T0	T
	S	BOOL	启动输入端	I，Q，M，D，L
	TV	S5TIME	定时时间（S5TIME 格式）	
	R	BOOL	复位输入端	
	Q	BOOL	定时器的状态	
	BI	WORD	当前时间（整数格式）	
	BCD	WORD	当前时间（BCD 码格式）	



断开延时定时器方框指令的示例如图 4−45 所示，本例时序图和运行结果与图 4−43 所示的是完全相同的，而且程序更为简洁。

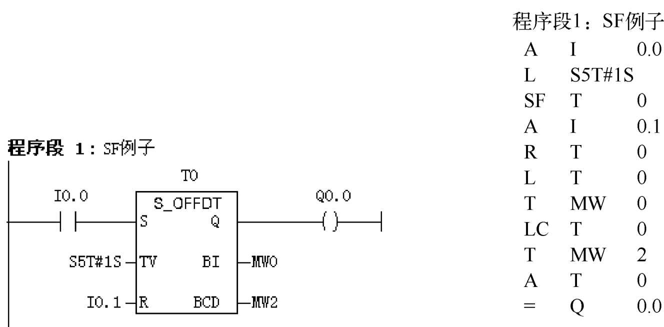


图 4−45　断开延时定时器方框指令的示例

4.4.2　计数器

计数器的功能是完成计数功能，可以实现加法计数和减法计数，计数范围是 0～999，计数器有 3 种类型：加计数器（S_CU）、减计数器（S_CD）和加减计数器（S_CUD）。

1．计数器的存储区

在 CPU 的存储区中，为计数器保留有存储区。该存储区为每个计数器地址保留一个 16 位的字。计数器的存储格式如图 4−46 所示，其中 BCD 码格式的计数值占用字的 0～11 位，共 12 位，而 12～15 位不使用；二进制格式的计数值占用字的 0～9 位，共 10 位，而 10～15 位不使用。梯形图指令支持 256 个计数器。

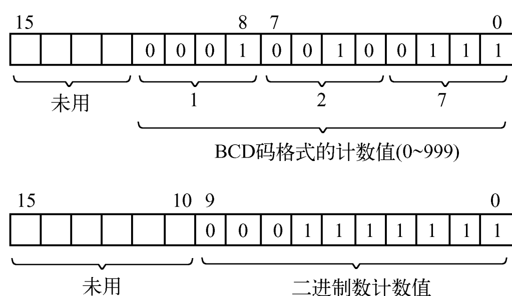


图 4−46　计数器字的格式

2．加计数器（S_CU）

加计数器（S_CU）在计数初始值预置输入端 S 上有上升沿时，PV 装入预置值，输入端 CU 每检测到一次上升沿，当前计数值 CV 加 1（前提是 CV 小于 999）；当前计数值大于 0 时，Q 输出为高电平“1”；当 R 端子的状态为“1”时，计数器复位，当前计数值 CV 为“0”，输出也为“0”。加计数器指令和参数见表 4−19。

表 4-19　加计数器指令和参数

LAD	参数	数据类型	说明	存储区
C no. S_CU CU　Q S PV　CV CV_BCD R	C no.	COUNTER	要启动的计数器号，如 C0	C
	CU	BOOL	加计数输入	I，Q，M，D，L
	S	BOOL	计数初始值预置输入端	
	PV	WORD	初始值的 BCD 码	
	R	BOOL	复位输入端	
	Q	BOOL	计数器的状态输出	
	CV	WORD	当前计数值（整数格式）	
	CV_BCD	WORD	当前计数值（BCD 码格式）	

用一个例子来说明加计数器指令的使用，梯形图和指令表如图 4-47 所示，与之对应的时序图如图 4-48 所示。当 I0.1 闭合时，MW20 将值赋给 PV（假设为 4）；当 I0.0 每产生一个上升沿，计数器 C0 计数 1 次，CV 加 1；只要计数值大于 0，Q0.0 输出高电平“1”。任何时候复位有效时，计数器 C0 复位，CV 清零，Q0.0 输出为低电平“0”。

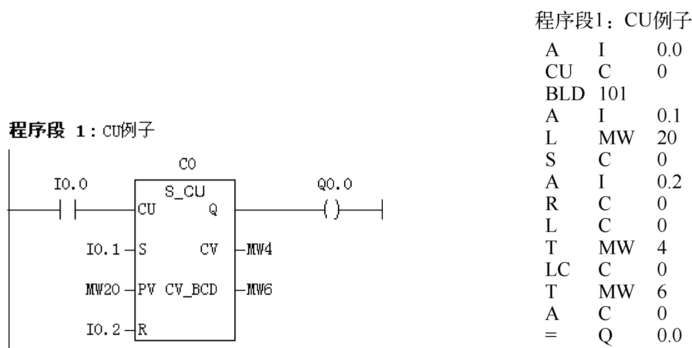


图 4-47　加计数器指令示例

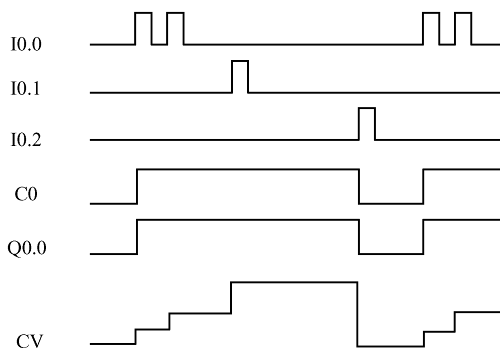


图 4-48　加计数器指令示例时序图

【关键点】S7-200 PLC 的增计数器（如 C0），当计数值到预置值时，C0 的常开触点闭合，常闭触点断开，S7-300 PLC 的计数器无此功能。

3．减计数器（S_CD）

减计数器（S_CD）在计数初始值预置输入端 S 上有上升沿时，PV 装入预置值，输入端

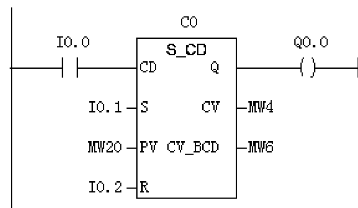


CD 每检测到一次上升沿，当前计数值 CV 减 1（前提是 CV 值大于 0），当 CV 等于 0 时，计数器的输出 Q 从状态“1”变成状态“0”；当 R 端子的状态为“1”时，计数器复位，当前计数值为“PV”，输出也为“0”。减计数器指令和参数见表 4-20。

表 4-20　减计数器指令和参数

LAD	参数	数据类型	说明	存储区
C no. S_CD CD　Q S PV　CV CV_BCD R	C no.	COUNTER	要启动的计数器号，如 C0	C
	CD	BOOL	减计数输入	I，Q，M，D，L
	S	BOOL	计数初始值预置输入端	
	PV	WORD	初始值的 BCD 码	
	R	BOOL	复位输入端	
	Q	BOOL	计数器的状态输出	
	CV	WORD	当前计数值（整数格式）	
	CV_BCD	WORD	当前计数值（BCD 码格式）	

用一个例子来说明减计数器指令的使用，梯形图和指令表如图 4-49 所示，与之对应的时序图如图 4-50 所示。当 I0.1 闭合时，MW20 将值赋给 PV（假设为 4），当 I0.0 每产生一个上升沿，计数器 C0 计数 1 次，CV 减 1，当 CV 值为 0 时，Q0.0 输出从“1”变成“0”。任何时候复位有效时，定时器 C0 复位，CV 值为 0，Q0.0 输出为低电平“0”。




```
程序段1：CD例子
A    I    0.0
CD   C    0
BLD  101
A    I    0.1
L    MW   20
S    C    0
A    I    0.2
R    C    0
L    C    0
T    MW   4
LC   C    0
T    MW   6
A    C    0
=    Q    0.0
```

图 4-49　减计数器指令示例

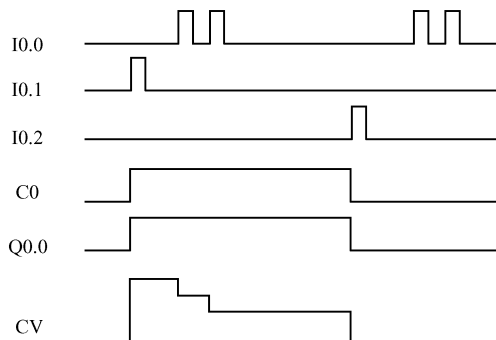


图 4-50　减计数器指令示例时序图

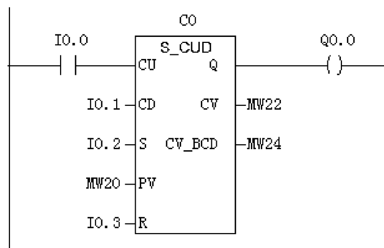
4．加-减计数器（S_CUD）

加-减计数器（S_CUD）在计数初始值预置输入端 S 上有上升沿时，PV 装入预置值，输入端 CD 每检测到一次上升沿，当前计数值 CV 减 1（前提是 CV 值大于 0）；输入端 CU 每检测到一次上升沿，当前计数值 CV 加 1（前提是 CV 值小于 999）；当 CD 和 CU 同时有上升沿时，CV 不变；计数值大于 0 时，计数器的输出 Q 从状态为“1”；计数值等于 0 时，计数器的输出 Q 从状态为“0”；当 R 端子的状态为“1”时，计数器复位，当前计数值为“0”，输出也为“0”。加-减计数器指令和参数见表 4-21。

表 4-21　加-减计数器指令和参数

LAD	参数	数据类型	说明	存储区
C no. S_CUD CU　Q CD S　CV PV　CV_BCD R	C no.	COUNTER	要启动的计数器号，如 C0	C
	CD	BOOL	减计数输入	I，Q，M，D，L
	CU	BOOL	加计数输入	
	S	BOOL	计数初始值预置输入端	
	PV	WORD	初始值的 BCD 码	
	R	BOOL	复位输入端	
	Q	BOOL	计数器的状态输出	
	CV	WORD	当前计数值（整数格式）	
	CV_BCD	WORD	当前计数值（BCD 码格式）	

用一个例子来说明加-减计数器指令的使用，梯形图和指令表如图 4-51 所示。当 I0.2 闭合时，MW20 将值赋给 PV（假设为 3），当 I0.1 每产生一个上升沿，计数器 C0 计数 1 次，CV 减 1，当 CV 值为 0 时，Q0.0 输出从“1”变成“0”；I0.0 是增计数端。任何时候复位有效时，定时器 C0 复位，CV 值为 0，Q0.0 输出为低电平“0”。



程序段1：CUD例子
```
A    I    0.0
CU   C    0
A    I    0.1
CD   C    0
A    I    0.2
L    MW   20
S    C    0
A    I    0.3
R    C    0
L    C    0
T    MW   22
LC   C    0
T    MW   24
A    C    0
=    Q    0.0
```

图 4-51　加-减计数器指令示例

4.5　数据处理与运算指令

4.5.1　装载与传送指令

装载 L 和传送指令 T 用于存储器之间或者存储区和过程输入、输出之间交换数据。装



载和传送指令需要累加器的参与。

装载（Load，L）指令将源操作数装入累加器 1，而累加器 1 原有的数据移入累加器 2。装入指令可以对字节（8 位）、字（16 位）、双字（32 位）数据并行操作。

传送（Transfer，T）指令将累加器 1 中的内容写入目的存储区中，累加器 1 的内容不变。

1．立即寻址的装载与传送指令

立即寻址的操作数直接在指令中，下面是使用立即寻址的例子。

```
L   −38                      //将 16 位十进制常数−38 装入累加器 1 的低字 ACCU1-L
L   L#5                      //将 32 位常数 5 装入累加器 1
L   2#0001_1001_1110_0010    //将 16 位二进制常数装入累加器 1 的低字 ACCU1-L
L   25.38                    //将 32 位浮点数常数（25.38）装入累加器 1
L   ‘ABCD’                   //将 4 个字符装入累加器 1
L   TOD#12:30:3.0            //将 32 位实时时间常数装入累加器 1
L   D#2004-2-3               //将 16 位日期常数装入累加器 1 的低字 ACCU1-L
L   C#50                     //将 16 位计数器常数装入累加器 1 的低字 ACCU1-L
L   T#1M20S                  //将 16 位定时器常数装入累加器 1 的低字 ACCU1-L
L   S5T#2S                   //将 16 位定时器常数装入累加器 1 的低字 ACCU1-L
L   P#M5.6                   //将指向 M5.6 的指针装入累加器 1
```

2．直接寻址的装载与传送指令

直接寻址在指令中直接给出存储器或寄存器的区域、长度和位置，例如用 MW200 指定位存储区中的字，地址为 200；下面是直接寻址的程序实例：

```
A   I0.0          //输入位 I0.0 的“与”（AND）操作
L   MB10          //将 8 位存储器字节装入累加器 1 最低的字节 ACCU1-LL
L   DIW15         //将 16 位背景数据字装入累加器 1 的低字 ACCU1-L
L   LD22          //将 32 位局域数据双字装入累加器 1
T   QB10          //将 ACCU1-LL 中的数据传送到过程映像输出字节 QB10
T   MW14          //将 ACCU1-L 中的数据传送到存储器字 MW14
T   DBD2          //将 ACCU1 中的数据传送到数据双字 DBD2
```

3．存储器间接寻址

在存储器间接寻址指令中，给出一个作地址指针的存储器，该存储器的内容是操作数所在存储单元的地址。在循环程序中经常使用存储器间接寻址。

地址指针可以是字或双字，如定时器（T）、计数器（C）、数据块（DB）、功能块（FB）和功能（FC）的编号范围小于 65535，使用字指针。其他地址则要使用双字指针，如果要用双字格式的指针访问一个字、字节或双字存储器，必须保证指针的位编号为 0，例如 P#Q20.0。

L　QB[DBD 10]　　//将输出字节装入累加器 1，输出字节的地址指针在数据双字 DBD10 中，如果 DBD10 的值为 2＃0000 0000 0000 0000 0000 0000 0010 0000，装入的是 QB4

A　M[LD 4]　　　//对存储器位作“与”运算，地址指针在数据双字 LD4 中，如果 LD4 的值为 2＃0000 0000 0000 0000 0000 0000 0010 0011，则是对 M4.3 进行操作

4．寄存器间接寻址

地址寄存器 AR1 和 AR2 的内容加上偏移量形成地址指针，指向数值所在的存储单元。其中第 0～2 位（xxx）为被寻址地址中位的编号（0～7），第 3～18 位为被寻址地址的字节编号（0～65535）。第 24～26 位（rrr）为被寻址地址的区域标识号，第 31 位 x = 0 为区域内的间接寻址，第 31 位 x = 1 为区域间的间接寻址。

第一种地址指针格式存储区的类型在指令中给出，例如 LDBB[AR1, P#6.0]。在某一存储区内寻址。第 24～26 位（rrr）应为 0。

第二种地址指针格式的第 24～26 位还包含存储区域标识符 rrr，区域间寄存器间接寻址。

如果要用寄存器指针访问一个字节、字或双字，必须保证指针中的位地址编号为 0。

指针常数＃P5.0 对应的二进制数为 2＃0000 0000 0000 0000 00000000 0010 1000。下面是区内间接寻址的例子：

```
L    P#5.0                  //将间接寻址的指针装入累加器 1
LAR1                        //将累加器 1 中的内容送到地址寄存器 1
A    M[AR1, P#2.3]          //AR1 中的 P#5.0 加偏移量 P#2.3, 实际上是对 M7.3 进行操作
=    Q[AR1, P#0.2]          //逻辑运算的结果送 Q5.2
L    DBW[AR1, P#18.0]       //将 DBW23 装入累加器 1
```

下面是区域间间接寻址的例子：

```
L    P#M6.0                 //将存储器位 M6.0 的双字指针装入累加器 1
LAR1                        //将累加器 1 中的内容送到地址寄存器 1
T    W[AR1, P#50.0]         //将累加器 1 中的内容传送到存储器字 MW56
```

P#M6.0 对应的二进制数为 2#1000 0011 0000 0000 0000 0000 0011 0000。因为地址指针 P#M6.0 中已经包含有区域信息，使用间接寻址的指令 T W[AR1, P#50]中没有必要再用地址标识符 M。

5．装载时间值或计数值

```
L     T5       //将定时器 T5 中的二进制时间值装入累加器 1 的低字中
LC    T5       //将定时器 T5 中的 BCD 码格式的时间值装入累加器 1 低字中
L     C3       //将计数器 C3 中的二进制计数值装入累加器 1 的低字中
LC    C16      //将计数器 C16 中的 BCD 码格式的值装入累加器 1 的低字中
```

6．地址寄存器的装载与传送指令

可以不经过累加器 1，与地址寄存器 AR1 和 AR2 交换数据。下面是应用实例：

```
LAR1   DBD20       //将数据双字 DBD20 中的指针装入 AR1
LAR2   LD180       //将局域数据双字 LD180 中的指针装入 AR2
LAR1   P#M10.2     //将带存储区标识符的 32 位指针常数装入 AR1
LAR2   P#24.0      //将不带存储区标识符 32 位指针常数装入 AR2
TAR1   DBD20       //AR1 中的内容传送到数据双字 DBD20
TAR2   MD24        //AR2 中的内容传送到存储器双字 MD24
```



7．装载与传送指令（MOVE）

对于初学者掌握指令表装载（L）与传送（T）是有些难度的，若读者先从梯形图中的传送指令学起，则容易理解，特别是对梯形图比较熟悉的读者更是如此。

当允许输入端的状态为“1”时，启动此指令，将 IN 端的数值输送到 OUT 端的目的地地址中，IN 和 OUT 有相同的信号状态，装载与传送指令（MOVE）的指令及参数见表 4-22。

表 4-22　装载与传送指令（MOVE）指令及参数

LAD	参数	数据类型	说明	存储区
MOVE EN　ENO IN　OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	OUT	所有长度为 8、16 或 32 位的基本数据类型	目的地地址	
	IN		源数据源	

用一个例子来说明装载与传送指令（MOVE）的使用，梯形图和指令表如图 4-52 所示，当 I0.0 闭合，MW20 中的数值（假设为 8），传送到目的地地址 MW22 中，结果是 MW20 和 MW22 中的数值都是 8。Q0.0 的状态与 I0.0 相同，也就是说，I0.0 闭合时，Q0.0 为“1”；I0.0 断开时，Q0.0 为“0”。

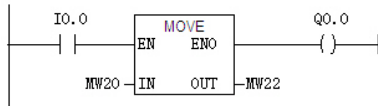


图 4-52　装载与传送梯形图指令示例

将图 4-52 所示的梯形图转化成指令表如下：

```
          A      I       0.0
          JNB    _001                  //如果 I1.0 = 0，则跳转到标号_001 处
          L      MW      20            //MW20 的值装入累加器 1 的低字
          T      MW      22            //累加器 1 低字的内容传送到 MW22
          SET                          //将 RLO 置为 1
          SAVE                         //将 RLO 保存到 BR 位
          CLR                          //将 RLO 置为 0
    _001: A      BR                    //状态字
          =      Q       0.0
```

【例 4-8】 用传送指令，设计一个梯形图将存储区 MB0~MB3 的数据清除。

【解】

MB0~MB3 实际上就是 MD0，因此用一条传送指令即可，梯形图如图 4-53 所示。

【关键点】传送指令的输入端的数据类型可以是常数、字节、整数、双整数和实数，使用非常灵活。

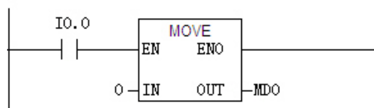


图 4-53　梯形图

4.5.2　比较指令

STEP 7 提供了丰富的比较指令，可以满足用户的各种需要。STEP 7 中的比较指令可以对下列数据类型的数值进行比较。

① 两个整数的比较（每个整数为 16 位）。

② 两个双整数的比较（每个双整数为 32 位）。

③ 两个实数的比较（每个实数为 32 位）。

【关键点】一个整数和一个双整数是不能直接进行比较的，因为它们之间的数据类型不同。一般先将整数转换成双整数，再对两个双整数进行比较。

比较指令有等于（EQ）、不等于（NQ）、大于（GT）、小于（LQ）、大于或等于（GE）和小于或等于（LE）。比较指令对输入 IN1 和 IN2 进行比较，如果比较结果为真，则逻辑运算结果 RLO 为“1”，反之则为“0”。

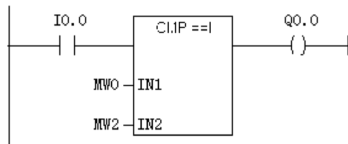
1．等于比较指令

等于指令有整数等于比较指令、双整数等于比较指令和实数等于比较指令 3 种。整数等于比较指令和参数见表 4-23。

表 4-23　整数等于比较指令和参数

LAD	参数	数据类型	说明	存储区
CMP ==I IN1 IN2	IN1	INT	比较的第一个数值	I、Q、M、D、L
	IN2	INT	比较的第二个数值	

用一个例子来说明整数等于比较指令，梯形图和指令表如图 4-54 所示。当 I0.0 闭合时，激活比较指令，MW0 中的整数和 MW2 中的整数比较，若两者相等，则 Q0.0 输出为“1”，若两者不相等，则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。




```
程序段1  等于比较
A     I     0.0
A(
L     MW    0
L     MW    2
==1
)
=     Q     0.0
```

图 4-54　整数等于比较指令示例



双整数等于比较指令和实数等于比较指令的使用方法与整数等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为双整数和实数。

2. 不等于比较指令

不等于比较指令有整数不等于比较指令、双整数不等于比较指令和实数不等于比较指令 3 种。整数不等于比较指令和参数见表 4-24。

表 4-24　整数不等于比较指令和参数

LAD	参数	数据类型	说明	存储区
CMP <>I IN1 IN2	IN1	INT	比较的第一个数值	I、Q、M、D、L
	IN2	INT	比较的第二个数值	

用一个例子来说明整数不等于比较指令，梯形图和指令表如图 4-55 所示。当 I0.0 闭合时，激活比较指令，MW0 中的整数和 MW2 中的整数比较，若两者不相等，则 Q0.0 输出为“1”，若两者相等，则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。

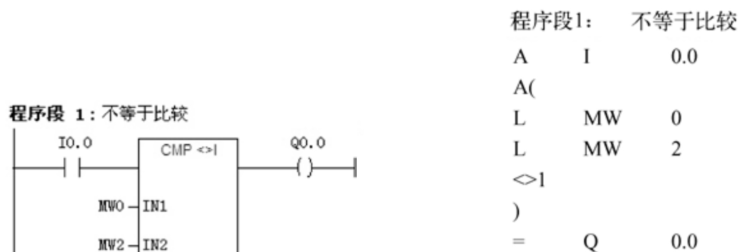


图 4-55　整数不等于比较指令示例

双整数不等于比较指令和实数不等于比较指令的使用方法与整数不等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为双整数和实数。使用比较指令的前提是数据类型必须相同。

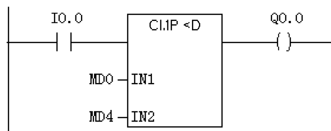
3. 小于比较指令

小于比较指令有整数小于比较指令、双整数小于比较指令和实数小于比较指令 3 种。双整数小于比较指令和参数见表 4-25。

表 4-25　双整数小于比较指令和参数

LAD	参数	数据类型	说明	存储区
CMP <D IN1 IN2	IN1	DINT	比较的第一个数值	I、Q、M、D、L
	IN2	DINT	比较的第二个数值	

用一个例子来说明双整数小于比较指令，梯形图和指令表如图 4-56 所示。当 I0.0 闭合时，激活双整数小于比较指令，MD0 中的双整数和 MD4 中的双整数比较，若前者小于后者，则 Q0.0 输出为“1”，否则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。




```
程序段1：小于比较
A    I     0.0
A(
L    MD    0
L    MD    4
<D
)
=    Q     0.0
```

图 4-56　双整数小于比较指令示例

整数小于比较指令和实数小于比较指令的使用方法与双整数小于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为整数和实数。使用比较指令的前提是数据类型必须相同。

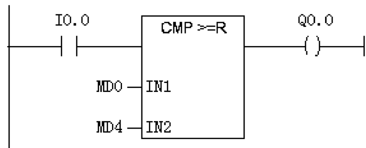
4．大于等于比较指令

大于等于比较指令有整数大于等于比较指令、双整数大于等于比较指令和实数大于等于比较指令 3 种。实数大于等于比较指令和参数见表 4-26。

表 4-26　实数大于等于比较指令和参数

LAD	参数	数据类型	说明	存储区
CMP >=R / IN1 / IN2	IN1	REAL	比较的第一个数值	I、Q、M、D、L
	IN2	REAL	比较的第二个数值	

用一个例子来说明实数大于等于比较指令，梯形图和指令表如图 4-57 所示。当 I0.0 闭合时，激活比较指令。MD0 中的实数和 MD4 中的实数比较，若前者大于或者等于后者，则 Q0.0 输出为“1”，否则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。




```
程序段：大等于比较
A    I     0.0
A(
L    MD    0
L    MD    4
>=R
)
=    Q     0.0
```

图 4-57　实数大于等于比较指令示例

整数大于等于比较指令和双整数大于等于比较指令的使用方法与实数大于等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为整数和双整数。使用比较指令的前提是数据类型必须相同。

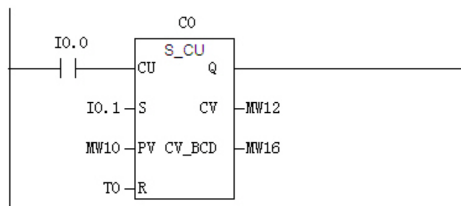
小于等于比较指令和小于比较指令类似，大于比较指令和大于等于比较指令类似，在此不再讲述小于等于比较指令和大于比较指令。

【例 4-9】 某设备上的控制器是 CPU 314C-2DP，设备上有一个光电传感器，检测工件，每检测到 1 只工件，计数一次，当计数到 3 只时，CPU 发出一个信号装箱，请设计梯形图。



【解】

梯形图如图 4-58 所示。光电传感器每检测一个工件时，计数器 C0 计 1 次数，当前计数值存放在 MW12 中，当计数 3 次时（MW12 中的数值大于等于 3），发出装箱信号 Q0.0，与此同时定时器 T0 开始定时，2s 后对计数器 C0 复位，重新计数。



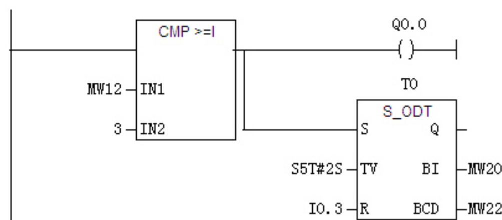


图 4-58　梯形图

4.5.3　转换指令

转换指令是将一种数据格式转换成另外一种格式进行存储。例如，要让一个整型数据和双整型数据进行算术运算，一般要将整型数据转换成双整型数据。

STEP 7 的转换指令见表 4-27。

表 4-27　转换指令

STL	LAD	说明
BTI	BCD_I	将累加器 1 中的 3 位 BCD 码转换成整数
ITB	I_BCD	将累加器 1 中的整数转换成 3 位 BCD 码
BTD	BCD_DI	将累加器 1 中的 7 位 BCD 码转换成双整数
DTB	DI_BCD	将累加器 1 中的双整数转换成 7 位 BCD 码
DTR	DI_R	将累加器 1 中的双整数转换成浮点数
ITD	I_DI	将累加器 1 中的整数转换成双整数
RND	ROUND	将浮点数转换为四舍五入的双整数
RND＋	CEIL	将浮点数转换为大于等于它的最小双整数
RND－	FLOOR	将浮点数转换为小于等于它的最大双整数
TRUNC	TRUNC	将浮点数转换为截位取整的双整数
CAW	－	交换累加器 1 低字中两个字节的位置
CAD	－	交换累加器 1 中 4 个字节的顺序

1．BCD转换成整数（BTI）

（1）BCD 码的格式

BCD 码是比较有用的，3 位格式如图 4-59 所示，二进制的 0～3 位是个位，4～7 位是十位，8～11 位是百位，11～15 位是符号位。7 位格式如图 4-60 所示，二进制的 0～3 位是个位，4～7 位是十位，8～11 位是百位，11～15 位是千位，16～19 位是万位，20～23 位是十万位，24～27 位是百万位，28～31 位是符号位。



图 4-59　3 位 BCD 码的格式

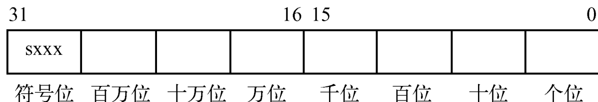


图 4-60　7 位 BCD 码的格式

（2）BCD 转换成整数指令（BTI）

BCD 转换成整数指令是将 IN 指定的内容以 BCD 码二～十进制格式读出，并将其转换为整数格式，输出到 OUT 端。如果 IN 端指定的内容超出 BCD 码的范围（例如 4 位二进制数出现 1010~1111 的几种组合），则执行指令时将会发生错误，使 CPU 进入 STOP 方式。BCD 转换成整数指令和参数见表 4-28。

表 4-28　BCD 转换成整数指令和参数

LAD	参数	数据类型	说明	存储区
BCD_I EN ENO IN OUT	EN	BOOL	使能（允许输入）	I、Q、M、D、L
	IN	WORD	输入的 BCD 数	
	ENO	BOOL	允许输出	
	OUT	INT	BCD 的整数	

用一个例子来说明 BCD 转换成整数指令，梯形图和指令表如图 4-61 所示。当 I0.0 闭合时，激活 BCD 转换成整数指令，IN 中的 BCD 数用 16 进制表示为 16#22（就是十进制的 22），转换完成后 OUT 端的 MW0 中的整数的十六进制是 16#16。



图 4-61　BCD 转换成整数指令示例

2．整数转换成BCD（ITB）

整数转换成 BCD 指令是将 IN 端指定的内容以整数的格式读入，然后将其转换为 BCD 码格式输出到 OUT 端。 如果 IN 端的整数大于 999，PLC 不停机，仍然正常运行。由于字的 BCD 码最大只能表示 C#999（最高 4 位为符号位）。若 IN 端的内容大于 999，CPU 将 IN 端的内容直接送到 OUT 端输出，不经过 I_BCD 的转换。这时 OUT 输出的内容可能超出 BCD 码的范围。另外 OUT 端的内容若为 BCD 码，也有可能是超过 999 的整数转换出来的，例如整数 2457 通过 I_BCD 指令以后，OUT 的值为 C#999。因此在使用 I_BCD 指令时



应该保证整数小于等于 999。此外，如果 IN 端的整数为负整数时。转换出的 BCD 码最高 4 位为“1”。整数转换成 BCD 指令和参数见表 4-29。

表 4-29　整数转换成 BCD 指令和参数

LAD	参数	数据类型	说明	存储区
I_BCD EN ENO IN OUT	EN	BOOL	使能（允许输入）	I、Q、M、D、L
	IN	INT	输入的整数	
	ENO	BOOL	允许输出	
	OUT	WORD	整数转化成的 BCD 数	

用一个例子来说明整数转换成 BCD 指令，梯形图和指令表如图 4-62 所示。当 I0.0 闭合时，激活整数转换成 BCD 指令，IN 中的整数存储在 MW0 中，（假设用 16 进制表示为 16#16），转换完成后 OUT 端的 MW2 中的 BCD 数是 22。

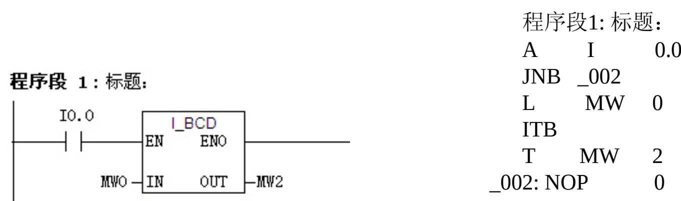


图 4-62　整数转换成 BCD 指令示例

3．整数转换成双整数（ITD）

整数转换成双整数指令是将 IN 端指定的内容以整数的格式读入，然后将其转换为双整数码格式输出到 OUT 端。整数转换成双整数指令和参数见表 4-30。

表 4-30　整数转换成双整数指令和参数

LAD	参数	数据类型	说明	存储区
I_DINT EN ENO IN OUT	EN	BOOL	使能（允许输入）	I、Q、M、D、L
	IN	INT	输入的整数	
	ENO	BOOL	允许输出	
	OUT	DINT	整数转化成的 BCD 数	

用一个例子来说明整数转换成双整数指令，梯形图和指令表如图 4-63 所示。当 I0.0 闭合时，激活整数转换成双整数指令，IN 中的整数存储在 MW0 中，（假设用 16 进制表示为 16#0016），转换完成后 OUT 端的 MD4 中的双整数是 16#0000 0016。

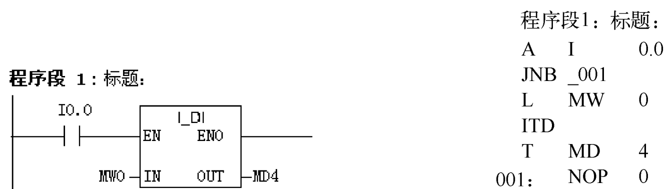


图 4-63　整数转换成双整数指令示例

4．双整数转换成实数（DTR）

双整数转换成实数指令是将 IN 端指定的内容以双整数的格式读入，然后将其转换为实数码格式输出到 OUT 端。实数格式在后续算术计算中是很常用的，如 3.14 就是实数形式。双整数转换成实数指令和参数见表 4-31。

表 4-31 双整数转换成实数指令和参数

LAD	参数	数据类型	说明	存储区
DI_REAL EN ENO IN OUT	EN	BOOL	使能（允许输入）	I、Q、M、D、L
	IN	DINT	输入的双整数	
	ENO	BOOL	允许输出	
	OUT	REAL	双整数转化成的实数	

用一个例子来说明双整数转换成实数指令，梯形图和指令表如图 4-64 所示。当 I0.0 闭合时，激活双整数转换成实数指令，IN 中的双整数存储在 MD0 中，（假设用 10 进制表示为 16），转换完成后 OUT 端的 MD4 中的实数是 16。一个实数要用 4 个字节存储。

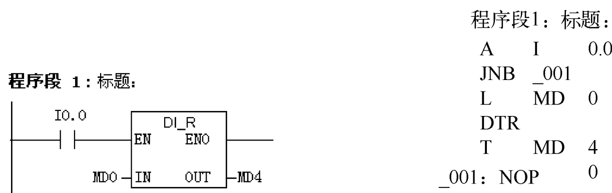


图 4-64 双整数转换成实数指令示例

【例 4-10】 请设计梯形图程序，将整数 16#22 转化成实数，并保存在 MD10 中。

【解】

STEP 7 中没有将整数直接转化成实数的指令，但可以通过数次转换将整数转换成实数，先将整数转换成双整数，再将双整数转换成实数。梯形图如图 4-65 所示。

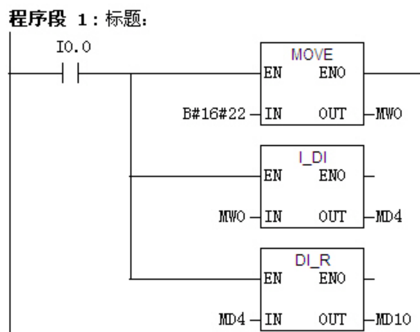


图 4-65 梯形图

5．实数四舍五入为双整数（ROUND）

ROUND 指令是将实数进行四舍五入取整后转换成双整数的格式。实数四舍五入为双整数指令和参数见表 4-32。



表 4-32　实数四舍五入为双整数指令和参数

LAD	参数	数据类型	说明	存储区
ROUND EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	实数（浮点型）	
	OUT	DINT	四舍五入后为双整数	

用一个例子来说明实数四舍五入为双整数指令，梯形图和指令表如图 4-66 所示。当 I0.0 闭合时，激活实数四舍五入指令，IN 中的实数存储在 MD0 中，假设这个实数为 3.14，进行四舍五入运算后 OUT 端的 MD4 中的双整数是 3，假设这个实数为 3.88，进行四舍五入运算后 OUT 端的 MD4 中的双整数是 4。

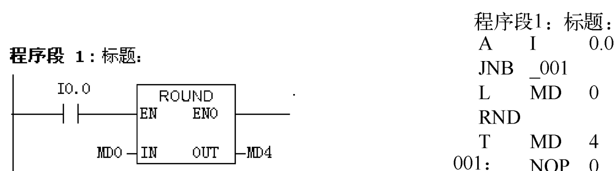


图 4-66　实数四舍五入为双整数指令示例

6．截取实数长整数部分（TRUNC）

截取实数长整数部分（TRUNC）指令是将实数舍弃小数部分，并将其转换为双整数格式。截取实数长整数部分（TRUNC）指令和参数见表 4-33。

表 4-33　截取实数长整数部分（TRUNC）指令和参数

LAD	参数	数据类型	说明	存储区
TRUNC EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	实数（浮点型）	
	OUT	DINT	转换后为双整数	

用一个例子来说明截取实数长整数部分（TRUNC）指令，梯形图和指令表如图 4-67 所示。当 I0.0 闭合时，激活截取实数长整数部分指令，IN 中的实数存储在 MD0 中，假设这个实数为 3.14，舍弃实数的小数部分后，OUT 端的 MD4 中的双整数是 3，假设这个实数为 3.88，舍弃实数的小数部分后 OUT 端的 MD4 中的双整数是 3。

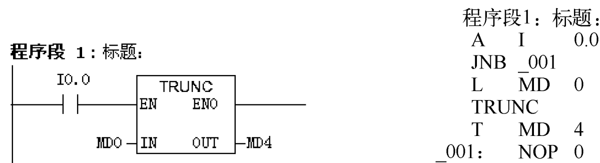


图 4-67　截取实数长整数部分（TRUNC）指令示例

【关键点】 TRUNC 和 ROUND 指令是有区别的，前者是将实数的小数部分舍弃，而后者是四舍五入。

7．向上取整（CEIL）

向上取整（CEIL）指令是将实数向上取整，结果为大于指定实数的最小双整数。向上取整（CEIL）指令和参数见表 4-34。

表 4-34　向上取整（CEIL）指令和参数

LAD	参数	数据类型	说明	存储区
CEIL EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	实数（浮点型）	
	OUT	DINT	转换后为双整数	

用一个例子来说明向上取整（CEIL）指令，梯形图和指令表如图 4-68 所示。当 I0.0 闭合时，激活向上取整指令，IN 中的实数存储在 MD0 中，假设这个实数为 3.14，向上取整后，OUT 端的 MD4 中的双整数是 4，假设这个实数为 3.88，向上取整后 OUT 端的 MD4 中的双整数是 4。

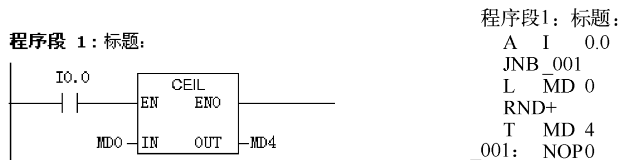


图 4-68　向上取整（CEIL）指令示例

8．向下取整（FLOOR）

向下取整（FLOOR）指令是将实数向下取整，结果为小于指定实数的最大双整数。向下取整（FLOOR）指令和参数见表 4-35。

表 4-35　向下取整（FLOOR）指令和参数

LAD	参数	数据类型	说明	存储区
CEIL EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	实数（浮点型）	
	OUT	DINT	转换后为双整数	

用一个例子来说明向下取整（FLOOR）指令，梯形图和指令表如图 4-69 所示。当 I0.0 闭合时，激活向下取整指令，IN 中的实数存储在 MD0 中，假设这个实数为 3.14，向下取整后，OUT 端的 MD4 中的双整数是 3，假设这个实数为–3.14，向下取整后 OUT 端的 MD4 中的双整数是–4。

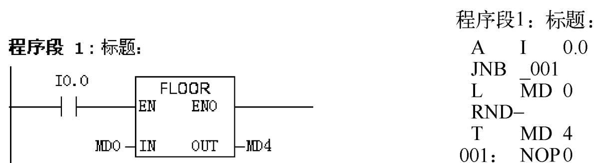


图 4-69　向下取整（FLOOR）指令示例



4.5.4　移位与循环指令

STEP 7 移位指令能将累加器的内容逐位向左或者向右移动。移动的位数由N决定。向左移N位相当于累加器的内容乘以 2^N，向右移相当于累加器的内容除以 2^N。移位指令在逻辑控制中使用也很方便。移位与循环指令见表 4-36。

表 4-36　移位与循环指令

名称	语句表	梯形图	描述
有符号整数右移	SSI	SHR_I	整数逐位右移，空出的位添上符号位
有符号双整数右移	SSD	SHR_DI	双整数逐位右移，空出的位添上符号位
16 位字左移	SLW	SHL_W	字逐位左移，空出的位添 0
16 位字右移	SRW	SHR_W	字逐位右移，空出的位添 0
16 位双字左移	SLD	SHL_DW	双字逐位左移，空出的位添 0
16 位双字右移	SRD	SHR_DW	双字逐位右移，空出的位添 0
双字循环左移	RLD	ROL_DW	双字循环左移
双字循环右移	RRD	ROR_DW	双字循环右移
双字＋CC1 循环左移	RLDA	—	双字通过 CC1（一共 33 位）循环左移
双字＋CC1 循环右移	RRDA	—	双字通过 CC1（一共 33 位）循环右移

1．字左移（SHL_W）

当字左移（SHL_W）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容送入累加器 1 低字中，并左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。字左移（SHL_W）指令和参数见表 4-37。

表 4-37　字左移（SHL_W）指令和参数

LAD	参数	数据类型	说明	存储区
SHL_W EN　ENO IN　OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	WORD	移位对象	
	N	WORD	移动的位数	
	OUT	WORD	移动操作的结果	

用一个例子来说明字左移指令，梯形图和指令表如图 4-70 所示。当 I0.0 闭合时，激活左移指令，IN 中的字存储在 MW0 中，假设这个数为 2#1001 1101 1111 1011，向左移 4 位后，OUT 端的 MW0 中的数是 2#1101 1111 1011 0000，字左移指令示意图如图 4-71 所示。

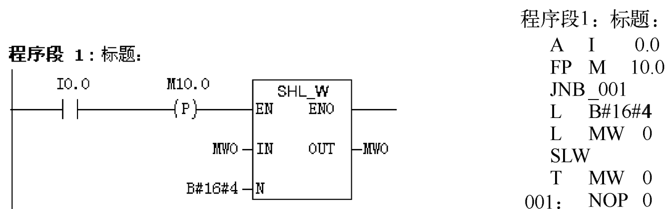


图 4-70　字左移指令示例

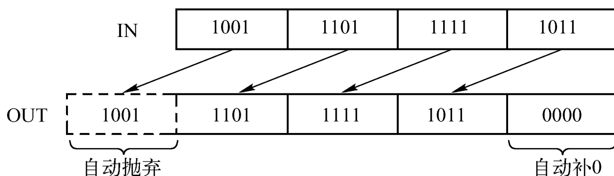


图 4-71　字左移指令示意图

【关键点】图 4-70 中的程序有一个上升沿，这样 I0.0 每闭合一次，左移 4 位，若没有上升沿，那么闭合一次，可能左移很多次。这点读者要特别注意。

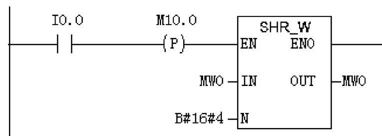
2．字右移（SHR_W）

当字右移（SHR_W）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指令的内容送入累加器 1 低字中，并右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。字右移（SHR_W）指令和参数见表 4-38。

表 4-38　字右移（SHR_W）指令和参数

LAD	参数	数据类型	说明	存储区
SHR_W EN　ENO IN　OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	WORD	移位对象	
	N	WORD	移动的位数	
	OUT	WORD	移动操作的结果	

用一个例子来说明字右移指令，梯形图和指令表如图 4-72 所示。当 I0.0 闭合时，激活右移指令，IN 中的字存储在 MW0 中，假设这个数为 2#1001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#0000 1001 1101 1111，字右移指令示意图如图 4-73 所示。



程序段1：标题：

```
      A     I     0.0
      FP    M     10.0
      JNB   _001
      L     B#16#4
      L     MW    0
      SRW
      T     MW    0
_001: NOP   0
```

图 4-72　字右移指令示例

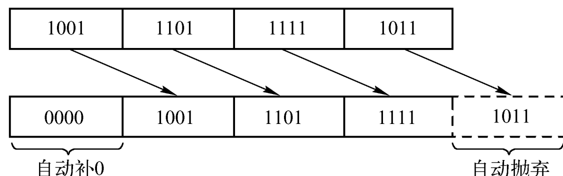


图 4-73　字右移指令示意图



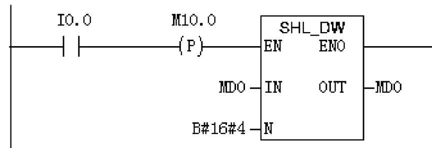
3．双字左移（SHL_DW）

当双字左移（SHL_DW）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字左移（SHL_DW）指令和参数见表 4-39。

表 4-39　双字左移（SHL_DW）指令和参数

LAD	参数	数据类型	说明	存储区
SHL_DW EN ENO IN OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	DWORD	移位对象	
	N	WORD	移动的位数	
	OUT	DWORD	移动操作的结果	

用一个例子来说明双字左移指令，梯形图和指令表如图 4-74 所示。当 I0.0 闭合时，激活双字左移指令，IN 中的双字存储在 MD0 中，假设这个数为 16#87654321，向左移 4 位后（半个字节），OUT 端的 MD0 中的数是 16#76543210。




```
程序段1：标题：
     A     I       0.0
     FP    M      10.0
     JNB  _001
     L     B#16#4
     L     MD     0
     SLD
     T     MD     0
_001：NOP           0
```

图 4-74　双字左移指令示例

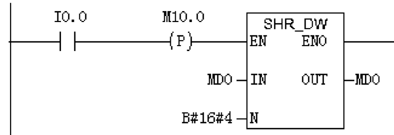
4．双字右移（SHR_DW）

当双字右移（SHR_DW）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字右移（SHR_DW）指令和参数见表 4-40。

表 4-40　双字右移（SHR_DW）指令和参数

LAD	参数	数据类型	说明	存储区
SHR_DW EN ENO IN OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	DWORD	移位对象	
	N	WORD	移动的位数	
	OUT	DWORD	移动操作的结果	

用一个例子来说明双字右移指令，梯形图和指令表如图 4-75 所示。当 I0.0 闭合时，激活双字右移指令，IN 中的双字存储在 MD0 中，假设这个数为 16#12345678，向左移 4 位后，OUT 端的 MD0 中的数是 16#01234567。




```
程序段1：标题：
     A    I      0.0
     FP   M    10.0
     JNB _001
     L     B#16#4
     L     MD  0
     SRD
     T     MD  0
_001:   NOP 0
```

图 4-75　双字右移指令示例

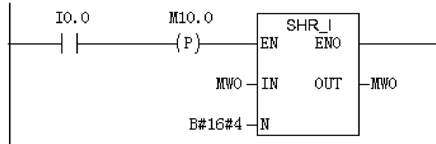
5．整数右移（SHR_I）

当整数右移（SHR_I）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中，与字的右移不同的是整数移位时，按照低位丢失、高位补符号位状态的原则，即正数高位补“0”，而负数补“1”。整数右移（SHR_I）指令和参数见表 4-41。

表 4-41　整数右移（SHR_I）指令和参数

LAD	参数	数据类型	说明	存储区
SHR_I EN　ENO IN　OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	WORD	移位对象	
	N	WORD	移动的位数	
	OUT	WORD	移动操作的结果	

用一个例子来说明整数右移指令，梯形图和指令表如图 4-76 所示。当 I0.0 闭合时，激活整数右移指令，IN 中的整数存储在 MW0 中，假设这个数为 2#0001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#0000 0001 1101 1111，而假设这个数为 2#1001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#1111 1001 1101 1111，其示意图如图 4-77 所示。




```
程序段1：标题：
     A    I      0.0
     FP   M    10.0
     JNB _001
     L     B#16#4
     L     MW  0
     SSI
     T     MW  0
_001:   NOP 0
```

图 4-76　整数右移指令示例

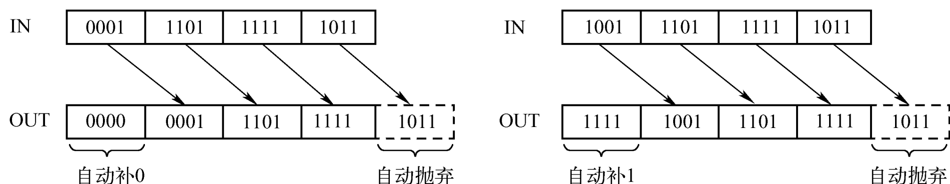


图 4-77　整数右移指令示意图



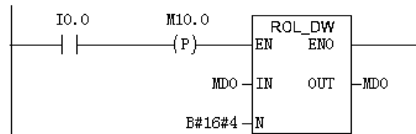
6．双字循环左移（ROL_DW）

当双字循环左移（ROL_DW）指令的 EN 位为高电平“1”时，将执行双字循环左位指令，将 IN 端指定的内容循环左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字循环左移（ROL_DW）指令和参数见表 4-42。

表 4-42　双字循环左移（ROL_DW）指令和参数

LAD	参数	数据类型	说明	存储区
ROL_DW EN ENO IN OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	DWORD	移位对象	
	N	WORD	移动的位数	
	OUT	DWORD	移动操作的结果	

用一个例子来说明双字循环左移（ROL_DW）指令的应用，梯形图和指令表如图 4-78 所示。当 I0.0 闭合时，激活双字循环左移指令，IN 中的双字存储在 MD0 中，假设这个数为 2#1001 1101 1111 1011 1001 1101 1111 1011，除最高 4 位外，其余各位向左移 4 位后，双字的最高 4 位，循环到双字的最低 4 位，结果是 OUT 端的 MD0 中的数是 2#1101 1111 1011 1001 1101 1111 1011 1001，其示意图如图 4-79 所示。



程序段1：标题：

```
      A     I      0.0
      FP    M      10.0
      JNB   _001
      L     B#16#4
      L     MD     0
      RLD
      T     MD     0
_001: NOP   0
```

图 4-78　双字循环左移指令示例

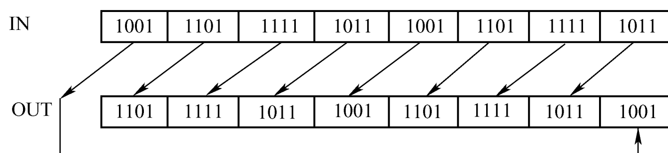


图 4-79　双字循环左移指令示意图

7．双字循环右移（ROR_DW）

当双字循环右移（ROR_DW）指令的 EN 位为高电平“1”时，将执行双字循环右移指令，将 IN 端指定的内容循环右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字循环右移（ROR_DW）指令和参数见表 4-43。

表 4-43　双字循环右移（ROR_DW）指令和参数

LAD	参数	数据类型	说明	存储区
ROR_DW EN ENO IN OUT N	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	DWORD	移位对象	
	N	WORD	移动的位数	
	OUT	DWORD	移动操作的结果	

用一个例子来说明双字循环右移（ROR_DW）指令的应用，梯形图和指令表如图 4-80 所示。当 I0.0 闭合时，激活双字循环右移指令，IN 中的双字存储在 MD0 中，假设这个数为 2#1001 1101 1111 1011 1001 1101 1111 1011，除最低 4 位外，其余各位向右移 4 位后，双字的最低 4 位，循环到双字的最高 4 位，结果是 OUT 端的 MD0 中的数是 2#1011 1001 1101 1111 1011 1001 1101 1111，其示意图如图 4-81 所示。

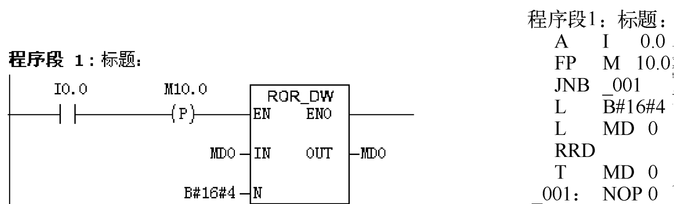


图 4-80　双字循环右移指令示例

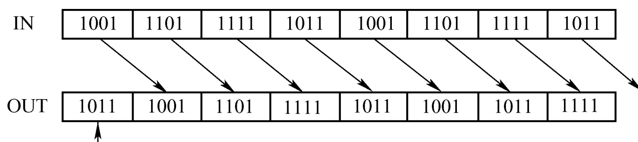


图 4-81　双字循环右移指令示意图

4.5.5　算术运算指令

算术运算指令非常重要，在模拟量的处理、PID 控制等很多场合都要用到算术运算。算术运算又分为整数算术运算和浮点数算术运算。

1．整数算术运算

整数算术运算又分为加法运算、减法运算、乘法运算和除法运算，其中每种运算方式又有整数型和双整数型两种。

（1）整数加（ADD_I）

当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的整数相加，结果送入 OUT 中。如果该结果超出了整数（16 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。整数加的表达式是：IN1＋IN2＝OUT。

整数加（ADD_I）指令和参数见表 4-44。



表 4-44 整数加（ADD_I）指令和参数

LAD	参数	数据类型	说明	存储区
ADD_I EN ENO IN1 IN2 OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	INT	相加的第 1 个值	
	IN2	INT	相加的第 2 个值	
	OUT	INT	相加的结果	

用一个例子来说明整数加（ADD_I）指令，梯形图和指令表如图 4-82 所示。当 I0.0 闭合时，激活整数加指令，IN1 中的整数存储在 MW0 中，假设这个数为 11，IN2 中的整数存储在 MW2 中，假设这个数为 21，整数相加的结果存储在 OUT 端的 MW4 中的数是 32。由于没有超出计算范围，所以 Q0.0 输出为“1”。假设 IN1 中的整数为 9999，IN2 中的整数为 30000，整数相加的结果存储在 OUT 端的 MW4 中的数是 25537。由于超出计算范围，所以 Q0.0 输出为“0”。

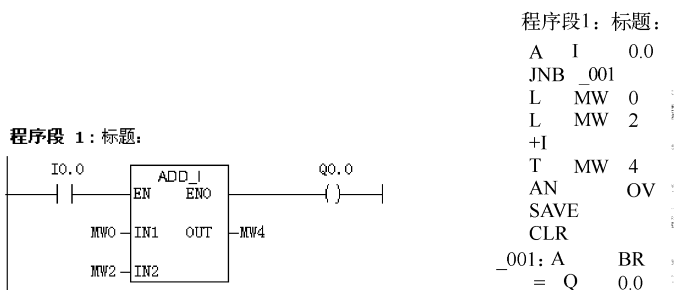


图 4-82 整数加（ADD_I）指令示例

双整数加（ADD_DI）指令与整数加（ADD_I）指令类似，只不过其数据类型为双整数，在此不赘述。

（2）双整数减（SUB_DI）

当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的双整数相减，结果送入 OUT 中。如果该结果超出了双整数（32 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。双整数减的表达式是：IN1－IN2＝OUT。

双整数减（SUB_DI）指令和参数见表 4-45。

表 4-45 双整数减（SUB_DI）指令和参数

LAD	参数	数据类型	说明	存储区
SUB_DI EN ENO IN1 IN2 OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	DINT	被减数	
	IN2	DINT	减数	
	OUT	DINT	差	

用一个例子来说明双整数减（SUB_DI）指令，梯形图和指令表如图 4-83 所示。当 I0.0 闭合时，激活双整数减指令，IN1 中的双整数存储在 MD0 中，假设这个数为 22，IN2 中的双整数存储在 MD4 中，假设这个数为 11，双整数相减的结果存储在 OUT 端的 MD4 中的数是 11。由于没有超出计算范围，所以 Q0.0 输出为“1”。

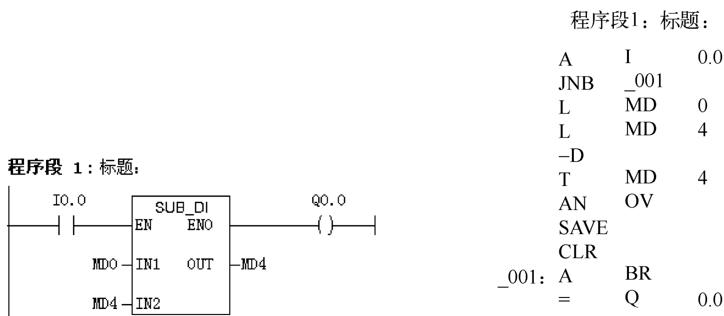


图 4-83　双整数减（SUB_DI）指令示例

整数减（SUB_I）指令与双整数减（SUB_DI）指令类似，只不过其数据类型为整数，在此不赘述。

（3）整数乘（MUL_I）

当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的整数相乘，结果送入 OUT 中。如果该结果超出了整数允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。整数乘的表达式是：IN1×IN2＝OUT。

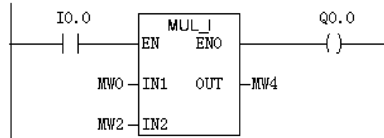
整数乘（MUL_I）指令和参数见表 4-46。

表 4-46　整数乘（MUL_I）指令和参数

LAD	参数	数据类型	说明	存储区
MUL_I EN ENO IN1 IN2 OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	INT	相乘的第 1 个值	
	IN2	INT	相乘的第 2 个值	
	OUT	INT	相乘的结果（积）	

用一个例子来说明整数乘（MUL_I）指令，梯形图和指令表如图 4-84 所示。当 I0.0 闭合时，激活整数乘指令，IN1 中的整数存储在 MW0 中，假设这个数为 11，IN2 中的整数存储在 MW2 中，假设这个数为 11，整数相乘的结果存储在 OUT 端的 MW4 中的数是 121。由于没有超出计算范围，所以 Q0.0 输出为“1”。假设 IN1 中的整数为 1000，IN2 中的整数为 1000，由于乘积超出范围，所以 Q0.0 输出为“0”。






```
程序段1：标题：
 A    I    0.0
 JNB  _001
 L    MW   0
 L    MW   2
 *I
 T    MW   4
 AN   OV
 SAVE
 CLR
_001： A    BR
 =    Q    0.0
```

图 4-84　整数乘（MUL_I）指令示例

双整数乘（MUL_DI）指令与整数乘（MUL_I）指令类似，只不过其数据类型为双整数，在此不赘述。

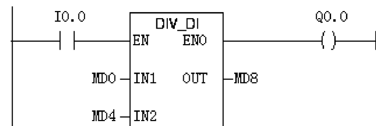
（4）双整数除（DIV_DI）

当允许输入端 EN 为高电平“1”时，输入端 IN1 中的双整数除以 IN2 中的双整数，结果送入 OUT 中。如果该结果超出了整数（32 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。双整数除（DIV_DI）指令和参数见表 4-47。

表 4-47　双整数除（DIV_DI）指令和参数

LAD	参数	数据类型	说明	存储区
DIV_DI EN　ENO IN1 IN2　OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	DINT	被除数	
	IN2	DINT	除数	
	OUT	DINT	除法的双整数结果（商）	

用一个例子来说明双整数除（DIV_DI）指令，梯形图和指令表如图 4-85 所示。当 I0.0 闭合时，激活双整数除指令，IN1 中的双整数存储在 MD0 中，假设这个数为 11，IN2 中的双整数存储在 MD4 中，假设这个数为 2，双整数相除的结果存储在 OUT 端的 MD8 中的数是 5，不产生余数。由于没有超出计算范围，所以 Q0.0 输出为“1”。




```
程序段1：标题：
 A    I    0.0
 JNB  _001
 L    MD   0
 L    MD   4
 /D
 T    MD   8
 AN   OV
 SAVE
 CLR
_001：A         BR
 =    Q    0.0
```

图 4-85　双整数除（DIV_DI）指令示例

【关键点】双整数除法不产生余数。

整数除（DIV_I）指令与双整数除（DIV_DI）指令类似，只不过其数据类型为整数，在此不赘述。

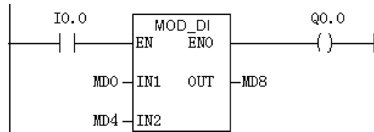
（5）返回双整数余数（MOD_DI）

当允许输入端 EN 为高电平“1”时，输入端 IN1 中的双整数除以 IN2 中的双整数，余数送入 OUT 中。IN1 和 IN2 中的数可以是常数。返回双整数余数指令和参数见表 4-48。

表 4-48　返回双整数余数（MOD_DI）指令和参数

LAD	参数	数据类型	说明	存储区
MOD_DI EN ENO IN1 IN2 OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	DINT	被除数	
	IN2	DINT	除数	
	OUT	DINT	除法的整数结果（商）	

用一个例子来说明返回双整数余数指令，梯形图和指令表如图 4-86 所示。当 I0.0 闭合时，激活返回双整数余数指令，IN1 中的整数存储在 MD0 中，假设这个数为 11，IN2 中的整数存储在 MD4 中，假设这个数为 2，双整数相除的余数存储在 OUT 端的 MD8 中的数是 1。由于没有超出计算范围，所以 Q0.0 输出为“1”。




```
程序段1：标题：
  A    I    0.0
  JNB  _001
  L    MD   0
  L    MD   4
  MOD
  T    MD   8
  AN   OV
  SAVE
  CLR
_001: A    BR
  =    Q    0.0
```

图 4-86　返回双整数余数指令示例

2．浮点数运算

浮点数函数有浮点算术运算函数、三角函数函数和反三角函数、对数函数、幂函数和绝对值函数等。浮点算术函数又分为加法运算、减法运算、乘法运算和除法运算函数。浮点数运算函数见表 4-49。

表 4-49　浮点数运算函数

语句表	梯形图	描述
+R	ADD_R	将累加器 1，2 中的浮点数相加，浮点数运算结果在累加器 1 中
- R	SUB_R	累加器 2 中的浮点数减去累加器 1 中的浮点数，运算结果在累加器 1 中
*R	MUL_R	将累加器 1，2 中的浮点数相乘，浮点数乘积在累加器 1 中
/R	DIV_R	累加器 2 中的浮点数除以累加器 1 中的浮点数，商在累加器 1



（续）

语句表	梯形图	描述
ABS	ABS	取累加器 1 中的浮点数的绝对值
SQR	SQR	求浮点数的平方
SQRT	SQRT	求浮点数的平方根
EXP	EXP	求浮点数的自然指数
LN	LN	求浮点数的自然对数
SIN	SIN	求浮点数的正弦函数
COS	COS	求浮点数的余弦函数
TAN	TAN	求浮点数的正切函数
ASIN	ASIN	求浮点数的反正弦函数
ACOS	ACOS	求浮点数的反余弦函数
ATAN	ATAN	求浮点数的反正切函数

（1）实数加（ADD_R）

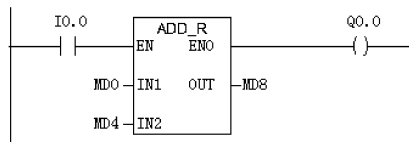
当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的实数相加，结果送入 OUT 中。如果该结果超出了允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。实数加的表达式是：IN1＋IN2＝OUT。

实数加（ADD_R）指令和参数见表 4-50。

表 4-50　实数加（ADD_R）指令和参数

LAD	参数	数据类型	说明	存储区
ADD_R EN ENO IN1 IN2 OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN1	REAL	相加的第 1 个值	
	IN2	REAL	相加的第 2 个值	
	OUT	REAL	相加的结果	

用一个例子来说明实数加（ADD_R）指令，梯形图和指令表如图 4-87 所示。当 I0.0 闭合时，激活实数加指令，IN1 中的实数存储在 MD0 中，假设这个数为 10.1，IN2 中的实数存储在 MD4 中，假设这个数为 21.1，实数相加的结果存储在 OUT 端的 MD8 中的数是 31.2。由于没有超出计算范围，所以 Q0.0 输出为“1”。



程序段1：标题：

```
    A    I     0.0
    JNB _001
    L    MD  0
    L    MD  4
    +R
    T    MD  8
    AN  OV
    SAVE
    CLR
_001:A       BR
    =   Q    0.0
```

图 4-87　实数加（ADD_R）指令示例

实数减（SUB_R）指令、实数乘（MUL_R）指令和实数除（DIV_R）指令的使用方法与前面的指令用法类似，在此不赘述。

【例 4-11】 将 53in 转换成以毫米为单位的整数，请设计梯形图。

【解】

1in=25.4mm，涉及实数乘法，先要将整数转换成双整数，再将双整数转化成实数，用实数乘法指令将英寸为单位的长度变为以毫米为单位的实数，最后四舍五入即可，梯形图如图 4-88 所示。

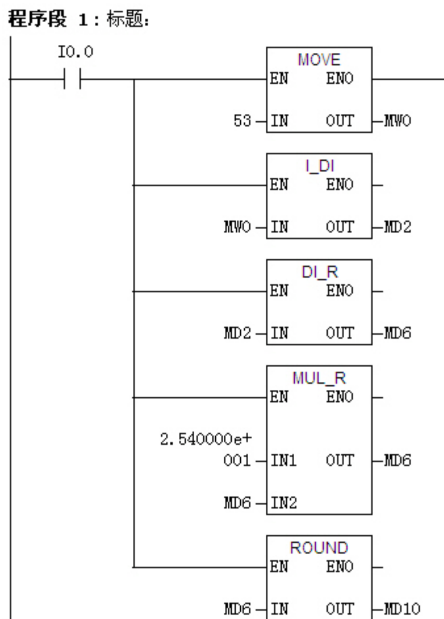


图 4-88　梯形图

（2）浮点数的绝对值（ABS）

当允许输入端 EN 为高电平“1”时，对输入端 IN 求绝对值，结果送入 OUT 中。IN 中的数可以是常数。浮点数的绝对值（ABS）的表达式是：OUT=｜IN｜。

浮点数的绝对值（ABS）指令和参数见表 4-51。

表 4-51　浮点数的绝对值（ABS）指令和参数

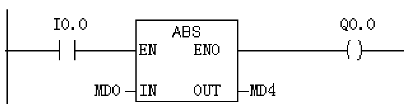
LAD	参数	数据类型	说明	存储区
ABS EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	输入值	
	OUT	REAL	输出值（绝对值）	

用一个例子来说明浮点数的绝对值（ABS）指令，梯形图和指令表如图 4-89 所示。当 I0.0 闭合时，激活浮点数的绝对值指令，IN 中的实数存储在 MD0 中，假设这个数为 10.1，



实数求绝对值的结果存储在 OUT 端的 MD4 中的数是 10.1，假设 IN 中的实数为－10.1，实数求绝对值的结果存储在 OUT 端的 MD4 中的数是 10.1。由于没有超出计算范围，所以 Q0.0 输出为“1”。

程序段 1：标题：



程序段1：标题：

```
      A    I    0.0
      JNB  _001
      L    MD   0
      ABS
      T    MD   4
      SET
      SAVE
      CLR
_001: A         BR
      =    Q    0.0
```

图 4-89　浮点数的绝对值（ABS）指令示例

（3）求正弦值（SIN）

当允许输入端 EN 为高电平“1”时，对输入端 IN 求正弦，结果送入 OUT 中。IN 中的数可以是常数。求正弦值（SIN）的表达式是：OUT＝SIN（IN）。

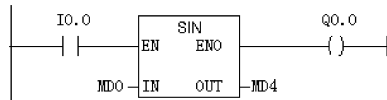
求正弦值（SIN）指令和参数见表 4-52。

表 4-52　求正弦值（SIN）指令和参数

LAD	参数	数据类型	说明	存储区
SIN EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	输入值	
	OUT	REAL	输出值（正弦值）	

用一个例子来说明求正弦值（SIN）指令，梯形图和指令表如图 4-90 所示。当 I0.0 闭合时，激活求正弦值指令，IN 中的实数存储在 MD0 中，假设这个数为 0.5，实数求正弦的结果存储在 OUT 端的 MD4 中的数是 0.479。由于没有超出计算范围，所以 Q0.0 输出为“1”。

程序段 1：标题：



程序段1：标题：

```
      A    I    0.0
      JNB  _001
      L    MD   0
      SIN
      T    MD   4
      AN   OV
      SAVE
      CLR
_001: A         BR
      =    Q    0.0
```

图 4-90　求正弦值（SIN）指令示例

【关键点】三角函数的输入值是弧度，而不是角度。

求余弦（COS）指令和求正切（TAN）指令的使用方法与前面的指令用法类似，在此不赘述。

（4）求反正弦值（ASIN）

当允许输入端 EN 为高电平“1”时，对输入端 IN 求反正弦，结果送入 OUT 中。IN 中的数可以是常数。求反正弦值（ASIN）的表达式是：OUT＝ ASIN（IN）。

求反正弦值（ASIN）指令和参数见表 4-53。

表 4-53　求反正弦值（ASIN）指令和参数

LAD	参数	数据类型	说明	存储区
ASIN EN ENO IN OUT	EN	BOOL	允许输入	I、Q、M、D、L
	ENO	BOOL	允许输出	
	IN	REAL	输入值	
	OUT	REAL	输出值（反正弦值）	

用一个例子来说明反正弦值（ASIN）指令，梯形图和指令表如图 4-91 所示。当 I0.0 闭合时，激活反正弦值指令，IN 中的实数存储在 MD0 中，假设这个数为 0.5，求实数反正弦值的结果存储在 OUT 端的 MD4 中的数是 0.524。由于没有超出计算范围，所以 Q0.0 输出为“1”。

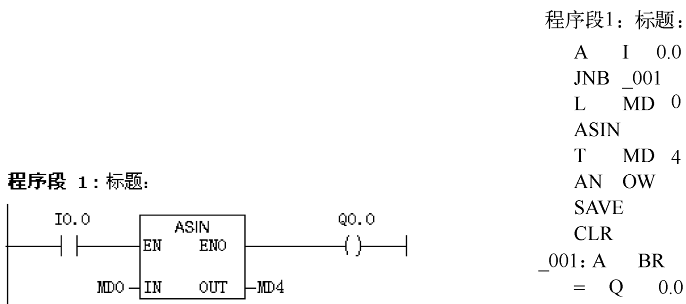


图 4-91　求反正弦值（ASIN）指令示例

【关键点】反正弦函数的输入值的范围是大于等于–1，而且小于等于 1。

求反余弦（ACOS）指令、求反正切（ATAN）指令的使用方法与前面的指令用法类似，在此不赘述。

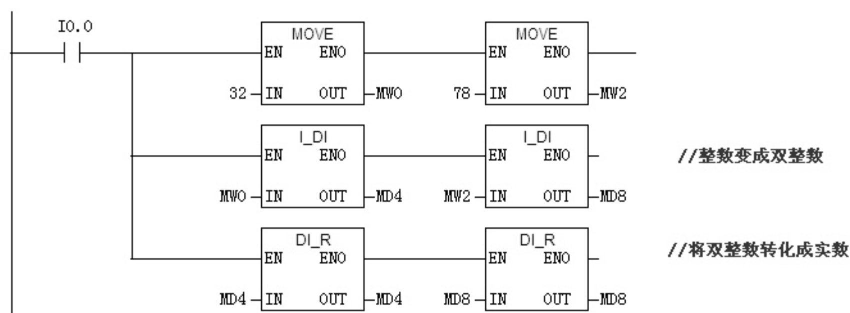
【例 4-12】 计算算式 $y = abs\left(\frac{\cos a - \sin b}{\pi}\right)$，其中 $a = 32^{o}, b = 78^{o}$ ，请设计梯形图。

【解】

三角函数要以弧度的形式表示，所以先将角度转换成弧度，再做实数运算，最后求绝对值，梯形图如图 4-92 所示。



程序段 1：标题：



程序段 2：标题：

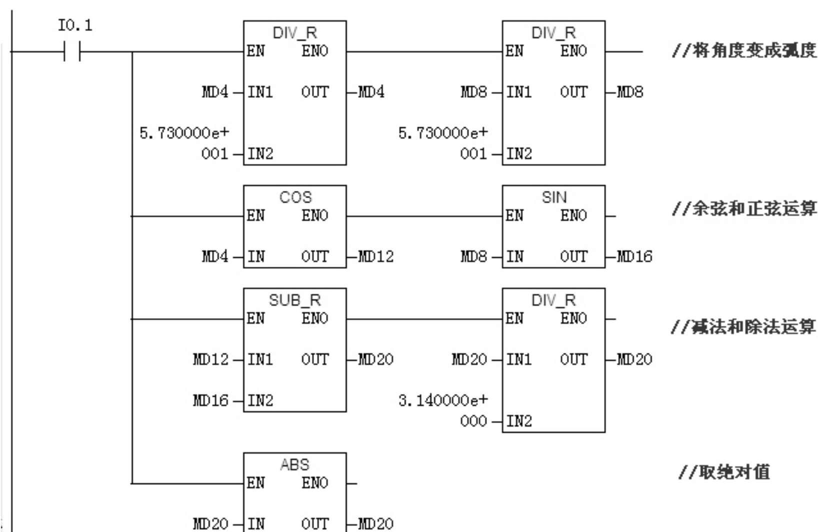


图 4-92　梯形图

4.5.6　控制指令

控制指令包括逻辑控制指令和程序控制指令。逻辑控制指令是指逻辑块中的跳转和循环指令。在没有执行跳转和循环指令之前，各语句按照先后顺序执行，也就是线型扫描。而逻辑控制指令终止了线型扫描，跳转到地址标号（Label）所指的地址，程序再次开始线型扫描。逻辑控制指令没有参数，只有一个地址标号，地址标号的作用如下：

（1）逻辑转移指令的地址是一个地址标号

地址标号最多由 4 个字母组成，第一个字符是字母，后面的字符可以是字母或者字符。

（2）目的地址标号必须从一个网络开始

跳转指令有几种形式，即无条件跳转、多分支跳转指令、与 RLO 和 BR 有关的跳转指令、与信号状态有关的跳转指令、与条件码 CC0 和 CC1 有关的跳转指令。逻辑控制指令见表 4-54。

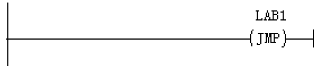
表 4-54　逻辑控制指令

指令	状态位触点指令	说明
JU	—	无条件跳转
JL	—	多分支跳转
JC	—	RLO=1 时跳转
JCN	—	RLO=0 时跳转
JCB	—	RLO=1 且 BR=1 时跳转
JNB	—	RLO=0 且 BR=1 时跳转
JBI	BR	BR=1 时跳转
JNBI	—	BR=0 时跳转
JO	OV	OV=1 时跳转
JOS	OS	OS=1 时跳转
JZ	==0	运算结果为 0 时跳转
JN	<> 0	运算结果非 0 时跳转
JP	> 0	运算结果为正时跳转
JM	< 0	运算结果为负时跳转
JPZ	>= 0	运算结果大于等于 0 时跳转
JMZ	<= 0	运算结果小于等于 0 时跳转
JUO	UO	指令出错时跳转
LOOP	—	循环指令

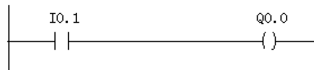
1．跳转指令

当逻辑位为“1”时，在块内执行跳转到标号处。跳转指令有两种使用情况：无条件跳转和条件跳转。当梯形图中的左母线与指令间没有其他梯形图元素时执行的是无条件跳转，示例如图 4-93 所示。

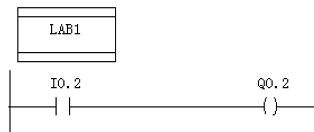
程序段 1：标题：



程序段 2：标题：



程序段 3：标题：




```
程序段1：标题：
JU      LAB1
程序段2：标题：
 A      I    0.1
 =      Q    0.0
程序段3：标题：
LAB1：A   I   0.2
      =   Q   0.2
```

图 4-93　无条件跳转指令示例

当逻辑运算的 RLO 为“1”时，执行的是条件跳转，示例如图 4-94 所示。当 I0.0 闭合时，执行条件跳转指令，跳转到标号 LAB1 处（本例为程序段 3），此时无论 I0.1 是否闭



合，Q0.0 都为低电平；而当 I0.0 断开时，不执行跳转指令，程序按照顺序执行，I0.1 闭合，Q0.0 为高电平，I0.1 断开，Q0.0 为低电平。

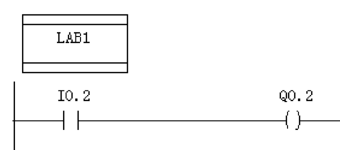
程序段 1：标题：



程序段 2：标题：



程序段 3：标题：




```
程序段1：标题：
 A      I      0.0
 JC     LAB1
程序段2：标题：
 A      I      0.1
 =      Q      0.0
程序段3：标题：
LAB1：A    I    0.2
      =    Q    0.2
```

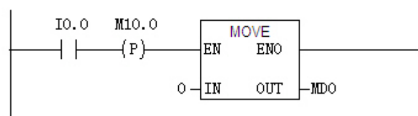
图 4-94　条件跳转指令示例

【例 4-13】 求和 $\sum_{i=0}^{100} i$，请设计梯形图。

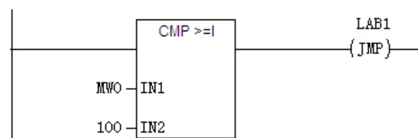
【解】

梯形图如图 4-95 所示。很显然要用跳转指令构造一个循环累加程序，当累加到 100 为止，运算结果保存在 MW8 中为 16#13BA（十进制为 5050）。

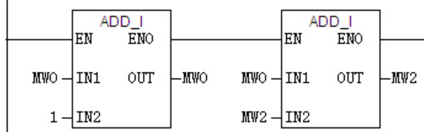
程序段 1：标题：



程序段 2：标题：



程序段 3：标题：



程序段 4：标题：

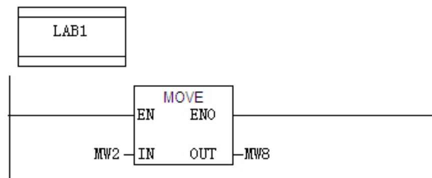


图 4-95　梯形图

2．若“否”则跳转指令

若“否”则跳转指令与跳转指令的使用方法类似，当逻辑位为“0”时，在块内执行跳转到标号处。示例如图 4-96 所示，当 I0.0 断开时，执行跳转指令，跳转到标号 LAB1 处（本例为程序段 3），此时无论 I0.1 是否闭合，Q0.0 都为低电平；而当 I0.0 闭合时，不执行跳转指令，程序按照顺序执行，I0.1 闭合，Q0.0 为高电平，I0.1 断开，Q0.0 为低电平。

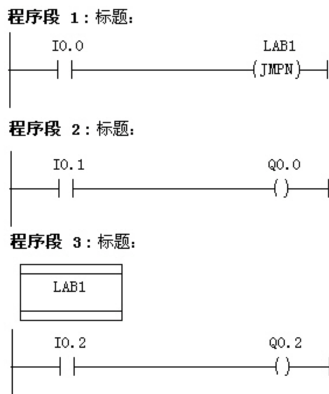


图 4-96　梯形图

4.6　S7-300/400 PLC的程序结构

4.6.1　功能、功能块和数据块

1．概述

在操作系统中包含了用户程序和系统程序，操作系统已经固化在 CPU 中，它提供 CPU 运行和调试的机制。CPU 的操作系统是按照事件驱动扫描用户程序的。用户程序写在不同的块中，CPU 按照执行的条件成立与否执行相应的程序块或者访问对应的数据块。用户程序则是为了完成特定的控制任务，是由用户编写的程序。用户程序通常包括组织块（OB）、功能块（FB）、功能（FC）和数据块（DB）。系统块包括系统功能（SFC）、系统功能块（SFB）和系统数据块（SDB）。

2．功能（FC）

（1）功能（FC）简介

1）功能（FC）是用户编写的程序块。功能是一种“不带内存”的逻辑块。属于 FC 的临时变量保存在本地数据堆栈中。执行 FC 时，该数据将丢失。为永久保存该数据，功能也可使用共享数据块。由于 FC 本身没有内存，因此，必须始终给它指定实际参数。不能给 FC 的本地数据分配初始值。

2）FC 里有一个局域变量表和块参数。局域变量表里有：IN（输入参数）、OUT（输出参数）、IN_OUT（输入/输出参数）、TEMP（临时数据）、RETURN（返回值 RET_VAL）。IN（输入参数）将数据传递到被调用的块中进行处理。OUT（输出参数）是将结果传递到调用

的块中。IN_OUT（输入/输出参数）将数据传递到被调用的块中，在被调用的块中处理数据后，再将被调用的块中发送的结果存储在相同的变量中。TEMP（临时数据）是块的本地数据，并且在处理块时将其存储在本地数据堆栈。关闭并完成处理后，临时数据就变得不再可访问。RETURN 包含返回值 RET_VAL。

（2）功能（FC）的应用

功能（FC）类似于 C 语言中的子程序，用户可以将具有相同控制过程的程序编写在 FC 中，然后在主程序 OB1 中调用。功能的应用并不复杂，先建立一个工程，再在管理器界面中选中“块”，接着单击菜单栏的 “插入”→“S7 块”→“功能”，即可插入一个空的功能。以下用两个例题讲解功能（FC）的应用。

【例 4-14】 用功能实现电动机的起停控制。

【解】

1）先新建一个工程，本例为“起停控制”。选中“块”，接着单击菜单栏的 “插入”→“S7 块”→“功能”，即可插入一个空的功能，如图 4-97 所示。



图 4-97 插入功能

2）如图 4-98 所示，在“属性－功能”界面中，输入功能的名称，再单击“确定”按钮。再双击“FC1”，打开功能，如图 4-99 所示。

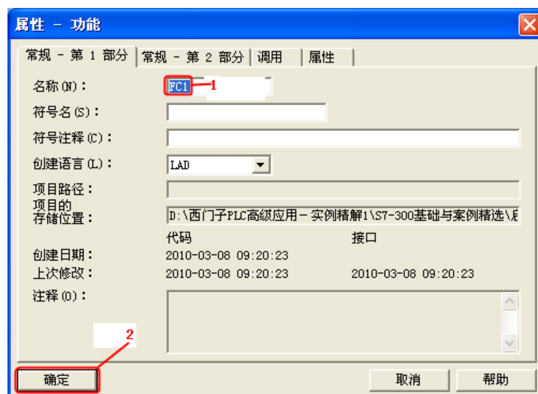


图 4-98 “属性－功能”界面

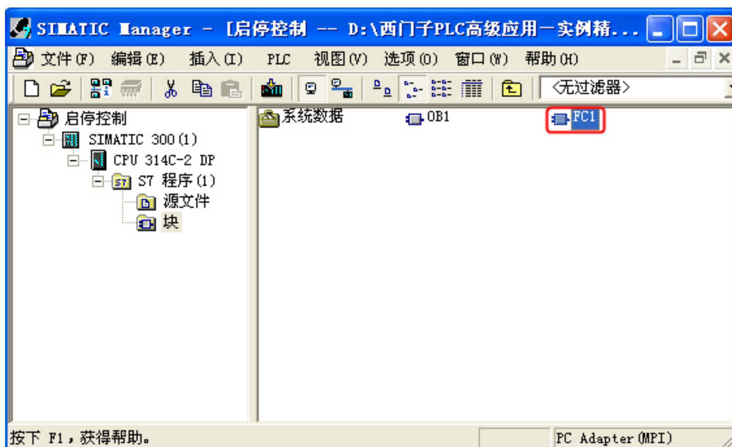


图 4-99　打开功能

3）在“程序编辑器”中输入如图 4-100 所示的程序，此程序能实现起停控制，再保存程序。

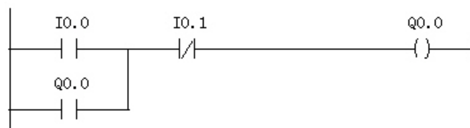


图 4-100　功能中的程序

4）回到管理器界面，双击“OB1”，打开主程序块“OB1”，如图 4-101 所示。

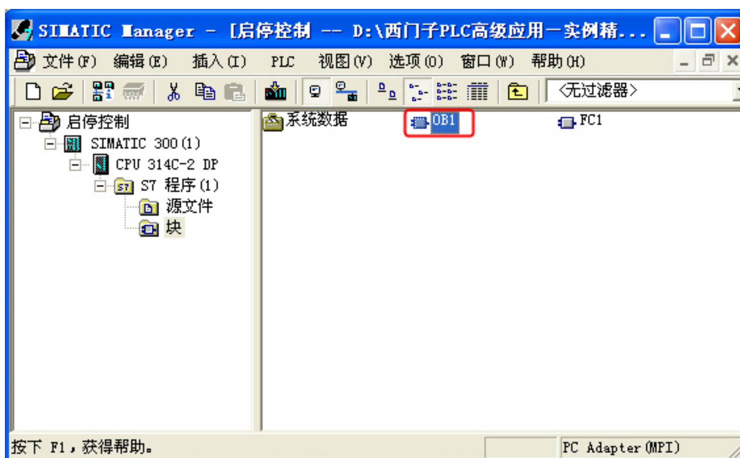


图 4-101　打开主程序块

5）将功能“FC1”拖入程序段 1，如图 4-102 所示。如果将整个工程下载到 PLC 中，就可以实现“起停控制”。



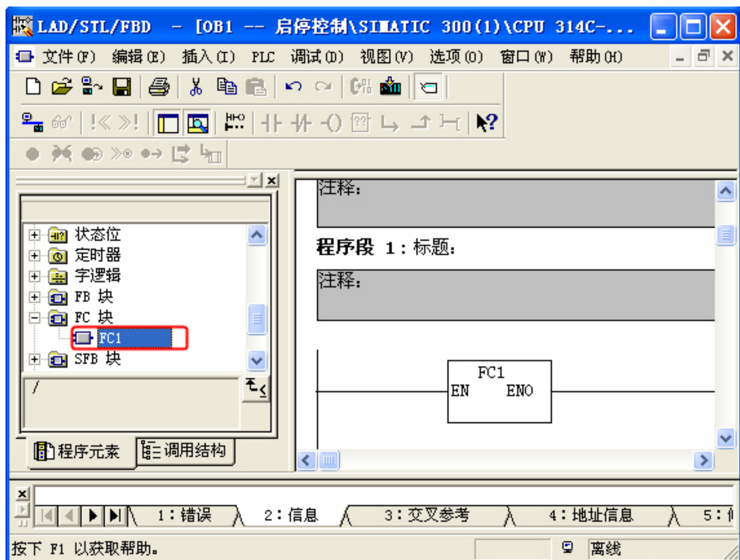


图 4-102　在主程序中调用功能

在例 4-14 中，只能用 I0.0 实现起动，而用 I0.1 实现停止，这种功能调用方式是绝对调用，显然不够方便，例 4-15 将用参数调用。

【例 4-15】 用功能实现电动机的起停控制。

【解】

本例的 1)、2) 步与例 4-14 相同，在此不再重复。

3) 在管理器中，双击功能块"FC1"，打开功能，弹出程序编辑器界面，先选中 IN（输入参数）新建参数"Start"和"Sto"，数据类型为"BOOL"，如图 4-103 所示。再选中 OUT（输出参数），新建参数"Motor"，数据类型为"BOOL"，如图 4-104 所示。最后在程序段 1 中输入程序，如图 4-104 所示，注意参数前都要加"#"。

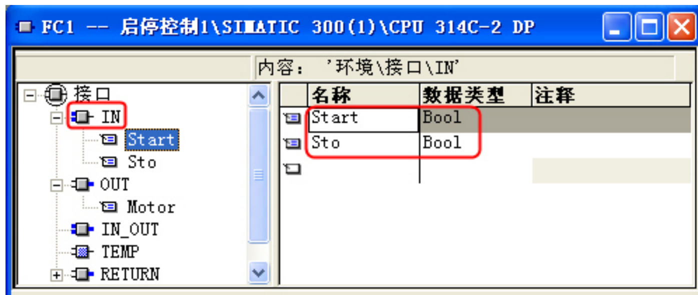


图 4-103　新建输入参数

4) 回到管理器界面，双击"OB1"，打开主程序块"OB1"，将功能"FC1"拖入程序段 1，如图 4-105 所示。如果将整个工程下载到 PLC 中，就可以实现"起停控制"。这个程序的功能"FC1"的调用比较灵活，与例 4-14 不同，起动只能是 I0.0，停止只能是 I0.1，在编写程序时，可以灵活应用。

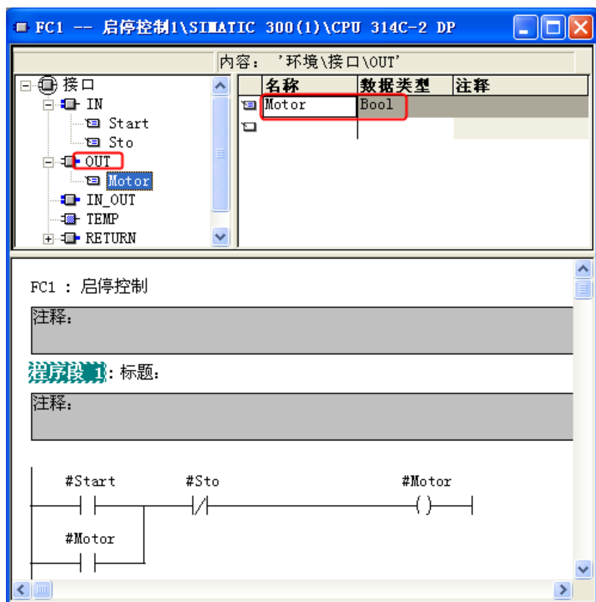


图 4-104　在主程序中调用功能

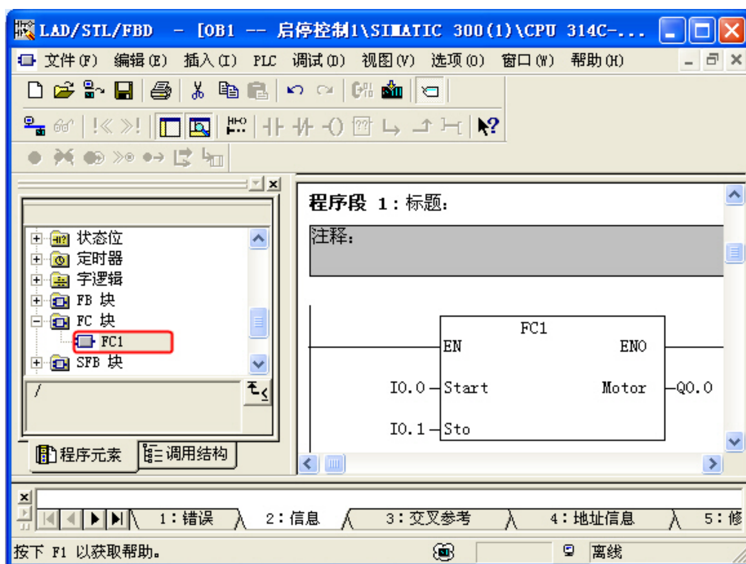


图 4-105　调用功能

3．功能块（FB）

功能块（FB）属于编程者自己编程的块。功能块是一种“带内存”的块。分配数据块作为其内存（实例数据块）。传送到 FB 的参数和静态变量保存在实例 DB 中。临时变量则保存在本地数据堆栈中。执行完 FB 时，不会丢失实例 DB 中保存的数据。但执行完 FB 时，会丢失保存在本地数据堆栈中的数据。

以下用一个例题来说明功能块的应用。



【例 4-16】 用功能块实现电动机的起停控制。

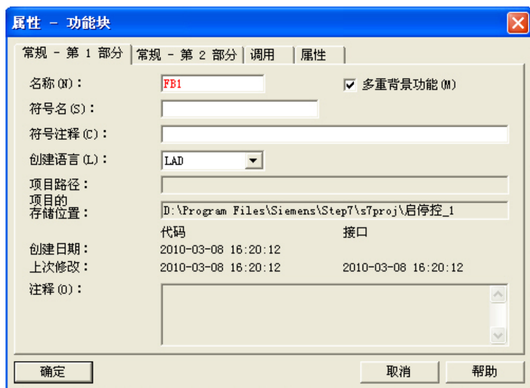
【解】

1）先新建一个工程，本例为“起停控制”，选中“块”，接着单击菜单栏的 “插入”→“S7 块”→“功能块”，即可插入一个空的功能块，如图 4-106 所示。



图 4-106　插入功能块

2）如图 4-107 所示，在“属性－功能块”界面中，输入功能块的名称，再单击“确定”按钮。再双击“FB1”，打开功能块，如图 4-108 所示。



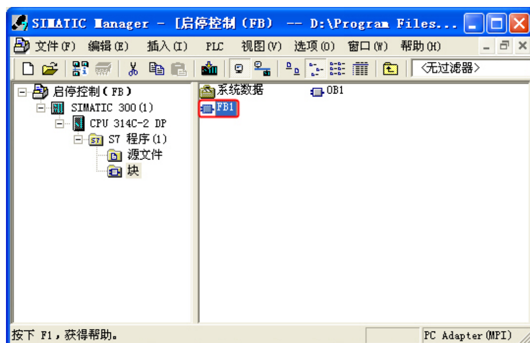


图 4-107　“属性－功能块”界面　　　　　　　　图 4-108　打开功能块

3）在“程序编辑器”中输入如图 4-109 所示的程序，此程序能实现起停控制，保存程序。

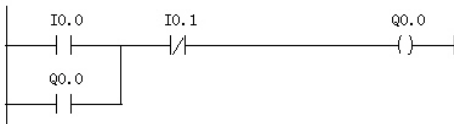


图 4-109　功能块中的程序

4）回到管理器界面，双击“OB1”，打开主程序块“OB1”，如图 4-110 所示。

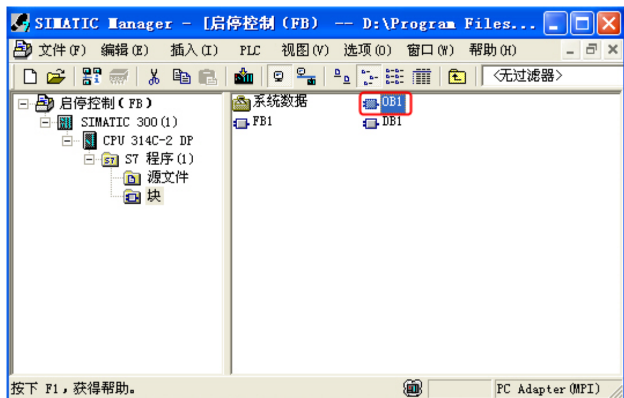


图 4-110　打开主程序块

5）将功能“FB1”拖入程序段 1，在 FB1 上输入数据块 DB1，如果这个数据块不存在，那么 STEP 7 可以提示读者建立它，如图 4-111 所示。将整个工程下载到 PLC 中，就可以实现“起停控制”。



图 4-111　调用功能块

【关键点】显而易见，功能块和功能的使用方法大致相同，只不过功能块需要背景数据块，这是两者的不同之处。

4．系统功能（SFC）

系统功能（SFC）是集成在 STEP 中，完成特定的功能。STEP 中有丰富的系统功能，供读者在编写程序时调用。以下用一个例子说明系统功能的使用方法。

【例 4-17】 用系统功能 SFC0 修改 CPU 314C-2DP 的系统时间。

【解】

1）先新建一个工程，本例为“时间”，选中“块”，双击“OB1”打开程序编辑器，如图 4-112 所示。



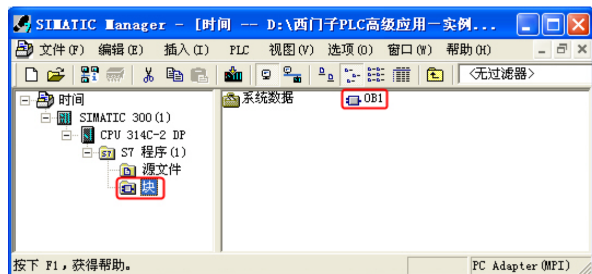


图 4-112　打开主程序块

2）将系统功能拖入程序编辑区。展开“库”→“System Function”，先选中“SFC0”，再将“SFC0”拖入到程序编辑区，如图 4-113 所示。

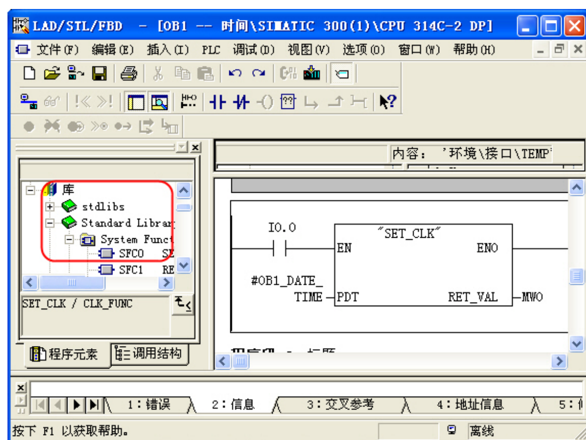
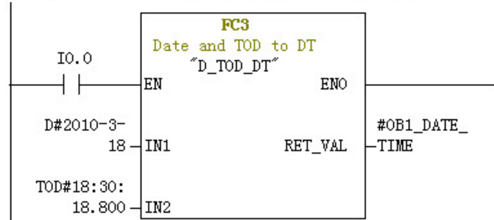


图 4-113　将系统功能拖入程序编辑区

3）编写如图 4-114 所示的程序，程序的作用是先将日期和时间合并，再用合并后的时间修改 CPU 的系统时间，修改后的系统时间是“2010-3-18-18:30:18.8”。

程序段 1：I0.0接通时，将日期和时间合并在一起



程序段 2：I0.1接通时，改写CPU的时钟

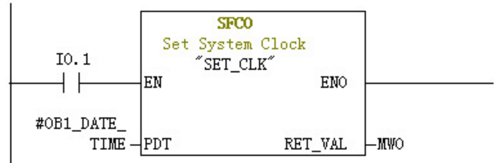


图 4-114　程序

系统功能块（SFB）的使用方法和系统功能（SFC）的使用方法类似，只不过系统功能块要用到背景数据块。

5．共享数据块（DB）

共享数据块（DB）与逻辑块不同，数据块不包含 STEP 7 指令。它们用来存储用户数据，即数据块包含用户程序使用的变量数据。共享数据块则用来存储可由所有其他块访问的用户数据。共享数据块（DB）的应用非常广泛。

以下用一个例题来说明数据块的应用。

【例 4-18】 用数据块实现电动机的起停控制。

【解】

1）先新建一个工程，本例为“数据块应用”，选中“块”，接着单击菜单栏的“插入”→“S7 块”→“数据块”，即可插入一个空的数据块，如图 4-115 所示。

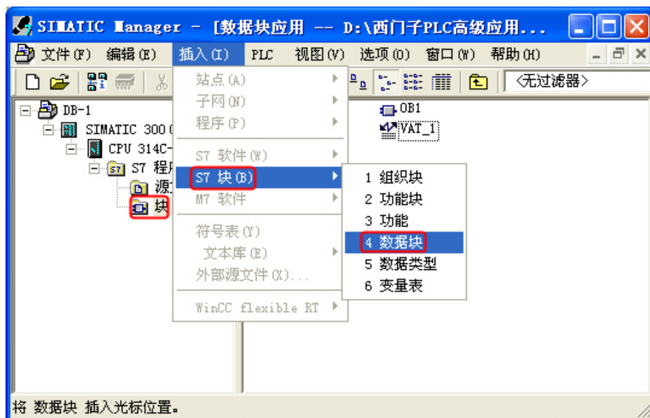


图 4-115　插入数据块

2）如图 4-116 所示，在“属性－数据块”界面中，输入数据块的名称，再单击“确定”按钮即可。

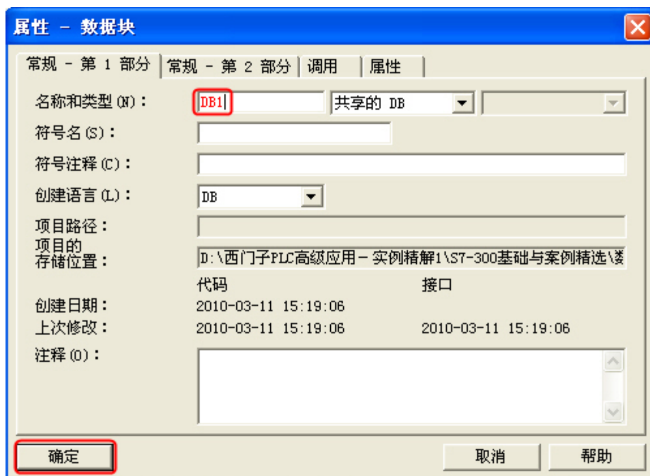


图 4-116　“属性－数据块”界面



3）回到管理器界面。选中“块”，接着单击菜单栏的“插入”→“S7 块”→“变量表”，即可插入一个空的变量表，如图 4-117 所示。

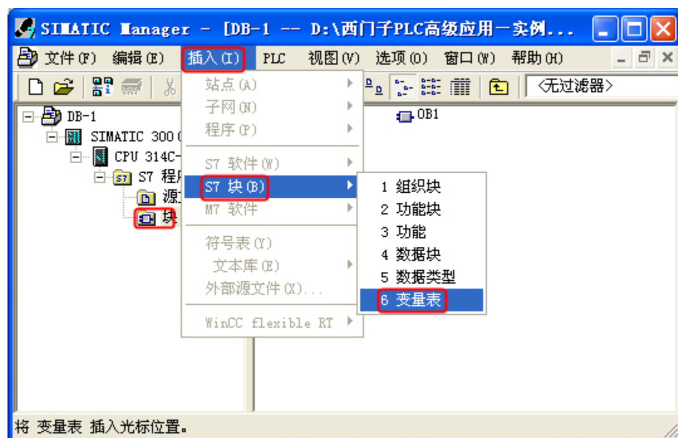


图 4-117　插入变量表

4）在“程序编辑器”中输入如图 4-118 所示的程序，此程序能实现起停控制，保存程序。



图 4-118　数据块中的程序

5）回到管理器界面，双击变量表“VAT_1”，打开变量表，并输入“1”处的地址、显示格式和修改数值，如图 4-119 所示。再将整个工程下载到 CPU 中，当单击“监视参数”和“修改变量”按钮时，Q0.0 闭合，可以控制电动机运行，当把“true”修改成“false”时，电动机停止运行。



图 4-119　监控参数

【关键点】数据块的使用比较灵活，除了上述的 BOOL 数据类型，还有其他数据类型，如 DB1.DBB0 表示字节，DB1.DBW0 表示字，DB1.DBD0 表示双字，在后面会用到。

4.6.2　组织块（OB）

组织块（OB）是操作系统与用户程序之间的接口。组织块由操作系统调用，控制循环中断驱动的程序执行、PLC 启动特性和错误处理。可以对组织块进行编程来确定 CPU 特性。

1．中断的概述

（1）中断过程

中断处理用来实现对特殊内部事件或外部事件的快速响应。CPU 检测到中断请求时，立即响应中断，调用中断源对应的中断程序（OB）。执行完中断程序后，返回被中断的程序。例如在执行主程序 OB1 块时，时间中断块 OB10 可以中断主程序块 OB1 正在执行的程序，转而执行中断程序块 OB10 中的程序，当中断程序块中的程序执行完成后，再转到主程序块 OB1 中，从断点处执行主程序。

中断源就是 I/O 模块的硬件中断、软件中断，例如日期时间中断、延时中断、循环中断和编程错误引起的中断。

（2）中断的优先级

执行一个组织块 OB 的调用可以中断另一个 OB 的执行。一个 OB 是否允许另一个 OB 中断取决于其优先级。OB 共有 29 个优先级，1 最低，29 最高。高优先级的 OB 可以中断低优先级的 OB。例如 OB10 的优先级是 2，而 OB1 的优先级是 1，所以 OB10 可以中断 OB1。背景 OB 的优先级最低。

优先级的顺序（后面的比前面的优先级高）：背景循环、主程序扫描循环、日期时间中断、时间延时中断、循环中断、硬件中断、多处理器中断、I/O 冗余错误、异步故障（OB80～87）、启动和 CPU 冗余，背景循环的优先级最低。

（3）对中断的控制

日期时间中断和延时中断有专用的允许处理中断和禁止中断的系统功能（SFC）。SFC 39“DIS_INT”用来禁止所有的中断、某些优先级范围的中断或指定的某个中断。SFC 40“EN_INT”用来激活（使能）新的中断和异步错误处理。如果用户希望忽略中断，可以下载一个只有块结束指令 BEU 的空 OB。

SFC 41“DIS_AIRT”延迟处理比当前优先级高的中断和异步错误。SFC 42“EN_AIRT”允许立即处理被 SFC 41 暂时禁止的中断和异步错误。

（4）组织块的分类

组织块只能由操作系统启动，它由变量声明表和用户编写的控制程序组成。

① 起动组织块 OB100～OB102。

② 循环执行的组织块。

③ 定期执行的组织块。

④ 事件驱动的组织块。

延时中断、硬件中断、异步错误中断 OB80～OB87，同步错误中断 OB121 和 OB122。

组织块的类型和优先级见表 4-55。

不是所有的中断组织块都能被 CPU 使用，不同类型的 CPU 可以调用的组织块一般不同，例如 CPU 314C-2DP 的循环中断仅能调用组织块 OB35，而不能调用 OB30～OB34 和 OB36～OB38 组织块。



表 4-55　组织块的类型和优先级

中断类型	组织块	优先级（默认）	启动事件
主程序扫描	OB1	1	用于循环程序处理的组织块（OB1）
时间中断	OB10 到 OB17	2	时间中断组织块（OB10～OB17）
延时中断	OB20	3	延时中断组织块（OB20～OB23）
	OB21	4	
	OB22	5	
	OB23	6	
循环中断	OB30	7	循环中断组织块（OB30～OB38）
	OB31	8	
	OB32	9	
	OB33	10	
	OB34	11	
	OB35	12	
	OB36	13	
	OB37	14	
	OB38	15	
硬件中断	OB40	16	硬件中断组织块（OB40～OB47）
	OB41	17	
	OB42	18	
	OB43	19	
	OB44	20	
	OB45	21	
	OB46	22	
	OB47	23	
DPV1 中断	OB 55	2	编程 DPV1 设备
	OB 56	2	
	OB 57	2	
多值计算中断	OB60 多值计算	25	多值计算-多个 CPU 的同步操作
同步循环中断	OB 61	25	组态 PROFIBUS DP 上的快速和等长过程响应时间
	OB 62	25	
	OB 63	25	
	OB 64	25	
冗余错误	OB70 I/O 冗余错误（仅在 H 系统中）	25	错误处理组织块（OB70～OB87 / OB121～OB122）
	OB72 CPU 冗余错误（仅在 H 系统中）	28	
异步错误	OB80 时间错误	25，在启动程序中出现异步错误 OB，那么为 28	错误处理组织块（OB70～OB87 / OB121～OB122）
	OB81 电源错误		
	OB82 诊断错误		
	OB83 插入/删除模块中断		
	OB84 CPU 硬件故障		

（续）

中断类型	组织块	优先级（默认）	启动事件
异步错误	OB 85 程序周期错误		
	OB86 机架故障		
	OB87 通信错误		
后台循环	OB90	29	后台组织块（OB90）
启动	OB100 重启动，（热重启动）	27	启动组织块（OB100/OB101/OB102）
	OB101 热重启动	27	
	OB102 冷重启动	27	
同步错误	OB121 编程错误	引起错误的 OB 的优先级	错误处理组织块（OB70～OB87 / OB121～OB122）
	OB122 访问错误		

2．主程序（OB1）

主程序（OB1）在前面经常用到，读者应该不会陌生。CPU 的操作系统定期执行 OB1。当操作系统完成启动后，将启动执行 OB1。在 OB1 中可以调用功能（FC）、系统功能（SFC）、功能块（FB）和系统功能块（SFB）。

执行 OB1 后，操作系统发送全局数据。重新启动 OB1 之前，操作系统将过程映像输出表写入输出模块中，更新过程映像输入表以及接受 CPU 的任何全局数据。

3．日期时钟中断组织块及其应用

CPU 可以使用的日期时间中断 OB 的个数与 CPU 的型号有关，例如 CPU 314C-2DP 只能用 OB10。

（1）指令简介

日期时钟中断组织块可以在某一特定的日期和时间执行一次，也可以从设定的日期时间开始，周期性地重复执行，例如每分钟、每小时、每天，每年执行一次。可以用 SFC28～SFC31 设置、取消、激活和查询日期时间中断。SFC28～SFC31 的参数见表 4-56。

表 4-56　SFC28～SFC31 的参数表

参数	声明	数据类型	存储区间	参数说明
OB_NR	INPUT	INT	I、Q、M、D、L、常数	OB 的编号
SDT	INPUT	DT	D、L、常数	启动日期和时间：将忽略指定的启动时间的秒和毫秒值 ，并将其设置为 0
PERIOD	INPUT	WORD	I、Q、M、D、L、常数	从启动点 SDT 开始的周期： W#16#0000 = 一次 W#16#0201 = 每分钟 W#16#0401 = 每小时 W#16#1001 = 每日 W#16#1202 = 每周 W#16#1401 = 每月 W#16#1801 = 每年 W#16#2001 = 月末
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果出错，则 RET_VAL 的实际参数将包含错误代码
STATUS	OUTPUT	WORD	I、Q、M、D、L	时间中断的状态



（2）日期时钟中断组织块的应用

以下用一个例题说明日期时钟中断组织块的应用。

【例 4-19】 从 2010 年 3 月 18 日 16 时起，每 1 小时中断一次，并将中断次数记录在一个存储器中。

【解】

一般有两种解法：

第一种解法比较简单，先打开 CPU 的属性界面，在“日期时钟中断”选项卡中，选择“激活”→“每小时”→“2010-3-18”→“16:00”，单击“确定”按钮，如图 4-120 所示。这个步骤的含义是：激活组织块 OB10 的中断功能，从 2010 年 3 月 18 日 16 时起，每 1 小时中断一次，再将组态完成的硬件下载到 CPU 中。



图 4-120　日期时钟中断

【关键点】初学者在使用此方法时，很容易忘记勾选“激活”或者不把组态的信息下载到 CPU 中去，请读者避免这样的失误。

打开 OB10，在程序编辑器中，输入程序如图 4-122 所示，运行的结果是从 2010 年 3 月 18 日 16 时起，每小时 MW2 中的数值增加 1，也就是记录了中断的次数。

第二种解法，并不需要在 CPU 的属性界面的“日期时钟中断”选项卡中进行设置，直接编写程序即可，但相对麻烦一些。主程序在 OB1 中，如图 4-121 所示，中断程序在 OB10 中，如图 4-122 所示。

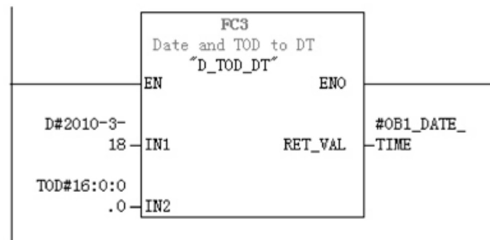
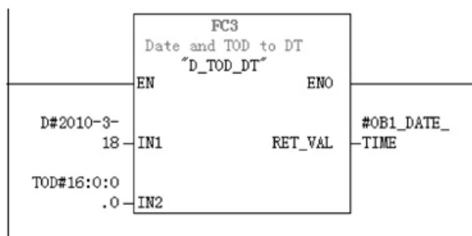
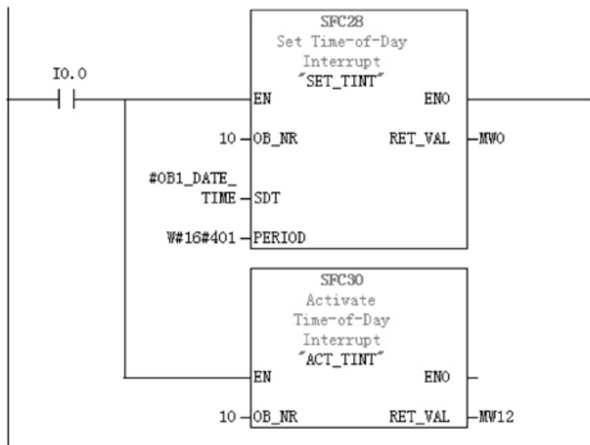


图 4-121　OB1 中的程序





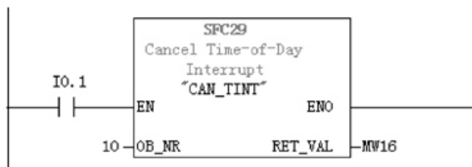


图 4-121　OB1 中的程序（续）

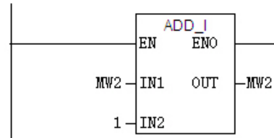


图 4-122　OB10 中的程序

4．循环中断组织块及其应用

CPU 可以使用的循环中断 OB 的个数与 CPU 的型号有关。所谓循环中断就是经过一段固定的时间间隔中断用户程序。

（1）循环中断指令

循环中断组织块是很常用的，STEP 7 中有 9 个循环中断组织块（OB30～OB38）。指令 SFC39～SFC42 来激活循环中断、禁止循环中断、禁用报警中断和启用报警中断。指令



SFC39～SFC42 的参数见表 4-57。

表 4-57　SFC39～SFC42 的参数表

参数	声明	数据类型	存储区间	参数说明
OB_NR	INPUT	INT	I、Q、M、D、L、常数	OB 的编号
MODE	INPUT	BYTE	I、Q、M、D、L、常数	指定禁用哪些中断和异步错误
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果出错，则 RET_VAL 的实际参数将包含错误代码

参数 MODE 指定禁用哪些中断和异步错误，含义比较复杂，MODE＝0 表示激活所有的中断和异步错误，MODE＝1 表示禁用所有新发生的和属于指定中断等级的事件，MODE＝2 表示禁用所有新发生的指定中断。具体可参考相关手册。

（2）循环中断组织块的应用

【例 4-20】 每隔 100ms 时间，CPU 314C-2DP 采集一次通道 0 上的数据。

【解】

很显然要使用循环组织块，有两种解法。

第一种解法比较简单，先打开 CPU 的属性界面，在“循环中断”选项卡中，将组织块 OB35 的执行时间定为“100ms”，单击“确定”按钮，如图 4-123 所示。这个步骤的含义是：设置组织块 OB35 的循环中断时间是 100ms，再将组态完成的硬件下载到 CPU 中。



图 4-123　设置循环中断

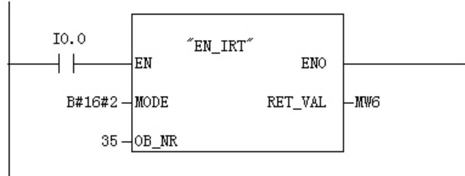
打开 OB35，在程序编辑器中，输入程序如图 4-125 所示，运行的结果是每 100ms 将通道 0 的采集到模拟量转化成数字量送到 MW0 中。

第二种解法，并不需要在 CPU 的属性界面的“循环中断”选项卡中进行设置，直接编写程序即可，但相对麻烦一些。主程序在 OB1 中，如图 4-124 所示，中断程序在 OB35 中，如图 4-125 所示。

5．硬件中断组织块及其应用

硬件中断组织块（OB40～OB47）用于快速响应信号模块（SM 输入/输出模块）、通信处理器（CP）和功能模块（FM）的信号变化。

程序段 1：激活循环中断，组织块为OB35



程序段 2：取消循环中断

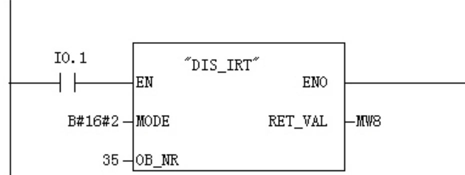


图 4-124　OB1 中的程序

程序段 1：每隔100ms将通道0的模拟量送入MW0

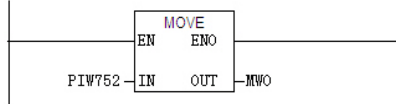


图 4-125　OB35 中的程序

硬件中断被模块触发后，操作系统将自动识别是哪一个槽的模块和模块中哪一个通道产生的硬件中断。硬件中断 OB 执行完后，将发送通道确认信号。

如果正在处理某一中断事件，又出现了同一模块同一通道产生的完全相同的中断事件，新的中断事件将丢失。

如果正在处理某一中断信号时同一模块中其他通道或其他模块产生了中断事件，当前已激活的硬件中断执行完后，再处理暂存的中断。

以下用一个例子说明硬件中断组织块的使用方法。

【例 4-21】 编写一段指令记录用户使用 I3.0 和 I3.1 按钮的次数，做成一个简单的“黑匣子”。

【解】

系统的硬件为 CPU 314C-2DP 和输入信号模块 SM321（Interrupt，带硬件中断功能）。先进行硬件组态，如图 4-126 所示，很明显信号输入模块的输入地址为“IB3”和“IB4”。双击“SM321 DI16XDC24V，Interrupt”，弹出信号模块的属性界面，如图 4-127 所示。在“输入选项卡”中，勾选“硬件中断”和“上升沿硬件中断发生器”（实际就是对 I3.0 和 I3.1 有效），最后单击“确定”按钮。

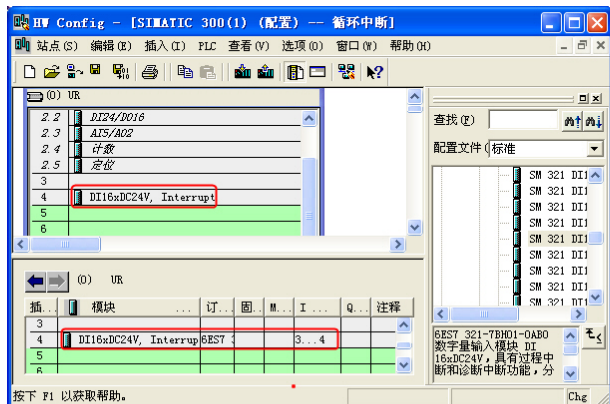


图 4-126　硬件组态界面



在组织块 OB40 中编写程序如图 4-128 所示，每次压下按钮，调用一次 OB40 中的程序一次，MW0 中的数值加 1，也就是记录了使用按钮的次数。

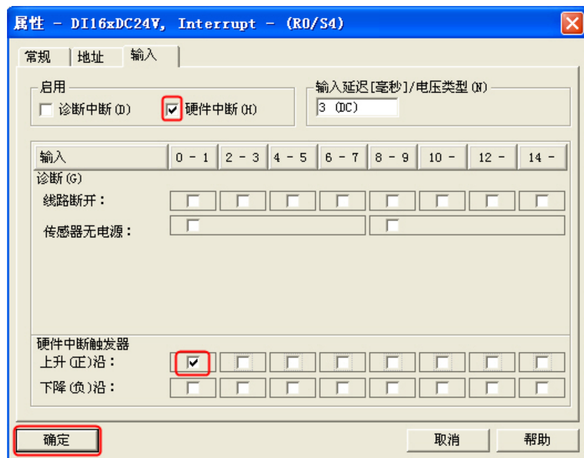


图 4-127　信号模块的属性界面

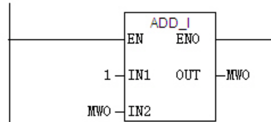


图 4-128　OB40 中的程序

【**关键点**】选用的输入模块“DI16XDC24V，Interrupt”必须具有硬件中断功能。

这个例子也可以用 SFC39 和 SFC40 来取消和激活中断。

6．错误组织块及其应用

（1）错误处理概述

S7-300/400 PLC 具有很强的错误（或称故障）检测和处理能力。PLC 内部的功能性错误或编程错误，而不是外部设备的故障。CPU 检测到错误后，操作系统调用对应的组织块，用户可以在组织块中编程，对发生的错误采取相应的措施。对于大多数错误，如果没有给组织块编程，出现错误时 CPU 将进入 STOP 模式。

（2）错误的分类

被 S7 CPU 检测到并且用户可以通过组织块对其进行处理的错误分为两个基本类型：

1）异步错误。其是与 PLC 的硬件或操作系统密切相关的错误，与程序执行无关，后果严重。异步错误 OB 具有最高等级的优先级，其他 OB 不能中断它们。同时有多个相同优先级的异步错误 OB 出现，将按出现的顺序处理。

2）同步错误（OB121 和 OB122）。是与程序执行有关的错误， 其 OB 的优先级与出现错误时被中断的块优先级相同，即同步错误 OB 中的程序可以访问块被中断时累加器和状态寄存器中的内容。对错误进行处理后，可以将处理结果返回被中断的块。

（3）时间错误处理组织块（OB80）

OB 执行时出现故障 S7-300 PLC CPU 的操作系统调用 OB80。这样的故障包括循环时间超出、执行 OB 时应答故障、向前移动时间以致于跃过了 OB 的启动的时间、CLR 后恢复 RUN 方式。

如果当循环中断 OB 仍在执行前一次调用时，该 OB 块的启动事件发生，操作系统调用 OB80。如果 OB80 未编程，CPU 变为 STOP 方式，可以使用 SFC39～SFC42 封锁或延时，或再使用时间故障 OB。

如果在同一个扫描周期中由于扫描时间超出 OB80 被调用两次，CPU 就变为 STOP 方式，可以通过在程序中适当的位置调用 SFC43“RE_TRIGR”来避免这种情况。

（4）电源故障处理组织块（OB81）

与电源（仅对 S7-400 PLC）或后备电池有关的故障事件发生时，S7-300 PLC CPU 的操作系统调用 OB81。如果 OB81 未编程，CPU 并不转换为 STOP 方式。可以使用 SFC39 至 42 来禁用、延时或再使用电源故障（OB81）。

（5）诊断中断处理组织块（OB82）

如果模块具有诊断能力又使能了诊断中断，当它检测到错误时，它输出一个诊断中断请求给 CPU，以及错误消失时，操作系统都会调用 OB82。当一个诊断中断被触发时，有问题的模块自动地在诊断中断 OB 的启动信息和诊断缓冲区中存入 4 个字节的诊断数据和模块的起始地址。可以用 SFC39～SFC42 来禁用、延时或再使用诊断中断（OB82），表 4-58 描述了诊断中断 OB82 的临时变量。

表 4-58　OB82 的变量申明表

变量	类型	描述
OB82_EV_CLASS	BYTE	事件级别和标识：B#16#38，离去事件；B#16#39，到来事件
OB82_FLT_ID	BYTE	故障代码
OB82_PRIORITY	BYTE	优先级：可通过 SETP 7 选择（硬件组态）
OB82_OB_NUMBR	BYTE	OB 号
OB82_RESERVED_1	BYTE	备用
OB82_IO_FLAG	BYTE	输入模板：B#16#54；输出模板：B#16#55
OB82_MDL_ADDR	WORD	故障发生处模板的逻辑起始地址
OB82_MDL_DEFECT	BOOL	模板故障
OB82_INT_FAULT	BOOL	内部故障
OB82_EXT_FAULT	BOOL	外部故障
OB82_PNT_INFO	BOOL	通道故障
OB82_EXT_VOLTAGE	BOOL	外部电压故障
OB82_FLD_CONNCTR	BOOL	前连接器未插入
OB82_NO_CONFIG	BOOL	模板未组态
OB82_CONFIG_ERR	BOOL	模板参数不正确
OB82_MDL_TYPE	BYTE	位 0～3：模板级别；位 4：通道信息存在；位 5：用户信息存在；位 6：来自替代的诊断中断；位 7：备用
OB82_SUB_MDL_ERR	BOOL	子模板丢失或有故障
OB82_COMM_FAULT	BOOL	通信问题
OB82_MDL_STOP	BOOL	操作方式（0：RUN，1：STOP）
OB82_WTCH_DOG_FLT	BOOL	看门狗定时器响应
OB82_RESERVED_2	BOOL	备用
OB82_RACK_FLT	BOOL	扩展机架故障
OB82_PROC_FLT	BOOL	处理器故障
OB82_EPROM_FLT	BOOL	EPROM 故障



（续）

变量	类型	描述
OB82_RAM_FLT	BOOL	RAM 故障
OB82_ADU_FLT	BOOL	ADC/DAC 故障
OB82_FUSE_FLT	BOOL	熔断器熔断
OB82_HW_INTR_FLT	BOOL	硬件中断丢失
OB82_RESERVED_3	BOOL	备用
OB82_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

在编写 OB82 的程序时，要从 OB82 的启动信息中获得与出现的错误有关的更确切的诊断信息。例如，其是哪一个通道出错，出现的是哪种错误。使用 SFC51“RDSYSST”也可以读出模块的诊断数据，用 SFC52“WR_USMSG”可以将这些信息存入诊断缓冲区。

现在通过结合模板的短线诊测应用和 SFC51 来说明诊断中断组织块 OB82 的使用方法。

首先，在 SIMATIC 管理器中新建一个项目，插入一个 300 站。再进行硬件组态，在机架上插入 CPU 314C-2DP 和一块具有中断功能模拟量输入模块 SM331，配置 SM331 模块的“输入（Inputs）”选项，选择 0-1 通道组为两线制电流（2DMU），其他通道组为电压，并注意模块的量程卡要与设置的相同。选中“启用（Enable）”框中的“诊断中断（Diagnostic Interrupt）”选项，选中“诊断（Diagnostics）”选项中的 0-1 通道组中的“组诊断（Group Diagnostics）”和“检查线路断开（with Check for Wire Break）”选项，如图 4-129 所示。



图 4-129　硬件组态

单击“确定”按钮，然后双击“CPU 314C-2DP”，选择“中断（Interrupts）”选项，可以看到 CPU 支持 OB82，如图 4-130 所示。硬件组态完成后，保存编译，下载到 CPU 中。

完成诊断程序。当在硬件组态中设定的诊断中断执行后，可以通过它的临时变量 OB82_MDL_ADDR 读出产生诊断中断的模块的逻辑地址。STEP 7 不能时时监控程序的运行。

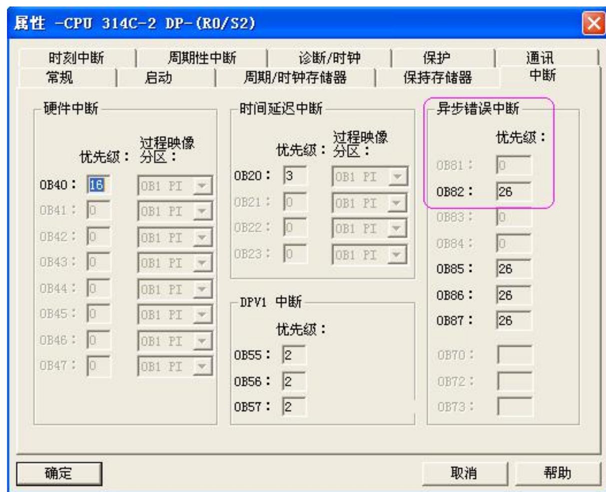


图 4-130　CPU 中的“中断（Interrupts）”选项

在 SIMATIC 管理器中 S7 Program（1）下插入一个 STL Source 文件 STL Source（1），如图 4-131 所示。

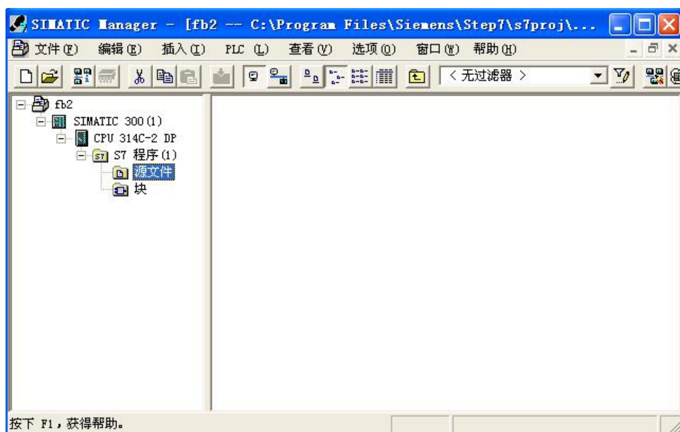


图 4-131　插入 STL Source 文件

打开 OB1，在“库（Libraries）”→“Standard Libraries”→“System Function Blocks”下找到 SFC51“RDSYSST DIAGNSTC”，按〈F1〉键，出现 SFC51 在线帮助信息，在帮助信息的最底部单击“Example for module diagnostics with the SFC51”，然后单击“STL Source File”，选中全部 STL Source 源程序复制到 STL Source（1）中，编译并保存。这是在 Blocks 中生成 OB1、OB82、DB13 和 SFC51。

打开 OB82，对其中的程序做简单的修改，将 19 和 20 行的程序复制到 go：后面，如图 4-132 所示。再进行保存，下载到 CPU 中。

下载完成后，将 CPU 上的模式选择开关切换到“RUN”状态，此时，CPU 上的“RUN”灯和“SF”灯会亮，SM331 模块上的“SF”灯也会亮。同时，查看 CPU 的诊断缓冲区可以获得相应的故障信息。





图 4-132　修改 OB82 中的程序

打开 DB13 数据块，在线监控，如图 4-133 所示。因为通道断线事件发生，所以诊断信息存储到 COME 数组中。

Address	Name	Type	Initial value	Actual value	Comment
0.0	COME[1]	BYTE	B#16#0	B#16#D	
1.0	COME[2]	BYTE	B#16#0	B#16#15	
2.0	COME[3]	BYTE	B#16#0	B#16#0	
3.0	COME[4]	BYTE	B#16#0	B#16#0	
4.0	COME[5]	BYTE	B#16#0	B#16#71	
5.0	COME[6]	BYTE	B#16#0	B#16#8	
6.0	COME[7]	BYTE	B#16#0	B#16#8	
7.0	COME[8]	BYTE	B#16#0	B#16#3	
8.0	COME[9]	BYTE	B#16#0	B#16#10	
9.0	COME[10]	BYTE	B#16#0	B#16#10	
10.0	COME[11]	BYTE	B#16#0	B#16#0	
11.0	COME[12]	BYTE	B#16#0	B#16#0	
12.0	COME[13]	BYTE	B#16#0	B#16#0	
13.0	COME[14]	BYTE	B#16#0	B#16#0	
14.0	COME[15]	BYTE	B#16#0	B#16#0	
15.0	COME[16]	BYTE	B#16#0	B#16#0	

图 4-133　DB13 中的数据变换

本例中 COME 数组字节的含义解释如下：

COME[1]=B#16#D：表示通道错误，外部故障和模块问题。

COME[2]=B#16#15：表示此段信息为模拟量模块的通道信息。

COME[3]=B#16#0：表示 CPU 处于运行状态，无字节 2 中标示的故障信息。

COME[4]=B#16#0：表示无字节 3 中标示的故障信息。

COME[5]=B#16#71：表示模拟量输入。

COME[6]=B#16#8：表示模块的每个通道有 8 个诊断位。

COME[7]=B#16#8：表示模块的通道数。

COME[8]=B#16#3：表示 0 通道错误和 1 通道错误，其他通道正常。

COME[9]=B#16#10：表示 0 通道断线。

COME[10]=B#16#10：表示 1 通道断线。

COME[11]=B#16#0：表示 2 通道正常，其他通道与 2 通道相同。

（6）插入/拔出模块中断组织块（OB83）

S7-400 PLC 可以在 RUN、STOP 或 STARTUP 模式下带电拔出和插入模块，但是不包括 CPU 模块、电源模块、接口模块和带适配器的 S5 模块，上述操作将会产生插入/拔出模块中断。

（7）CPU 硬件故障处理组织块（OB84）

当 CPU 检测到 MPI 网络的接口故障、通信总线的接口故障或分布式 I/O 网卡的接口故障时，操作系统调用 OB84。故障消除时也会调用该 OB 块。

（8）优先级错误处理组织块（OB85）

在以下情况下将会触发优先级错误中断：

① 产生了一个中断事件，但是对应的 OB 块没有下载到 CPU。

② 访问一个系统功能块的背景数据块时出错。

③ 刷新过程映像表时，I/O 访问出错，模块不存在或有故障。

（9）通信错误组织块（OB87）

在使用通信功能块或全局数据（GD）通信进行数据交换时，如果出现下列通信错误，操作系统将调用 OB87，有如下情况：

① 接受全局数据时，检测到不正确的帧标识符（ID）。

② 全局数据通信的状态信息数据块不存在或太短。

③ 接受到非法的全局数据包编号。

如果用于全局数据通信状态信息的数据块丢失，需要用 OB87 生成该数据块将它下载到 CPU。可以使用 SFC39～42 封锁或延时并使能通信错误 OB。

（10）同步错误组织块

同步错误是与执行用户程序有关的错误，OB121 用于对程序错误的处理，OB122 用于处理模块访问错误。

同步错误 OB 的优先级与检测到出错的块的优先级一致。

同步错误可以用 SFC 36“MASK_FLT”来屏蔽，用错误过滤器中的一位用来表示某种同步错误是否被屏蔽。错误过滤器分为程序错误过滤器和访问错误过滤器，分别占一个双字。屏蔽后的错误过滤器可以读出。

可以用 SFC 38“READ_ERR”读出已经发生的被屏蔽的错误。

① 编程错误组织块（OB121）。当有关程序处理的故障事件发生时，CPU 操作系统调用 OB121，OB121 与被中断的块在同一优先级中执行，表 4-59 描述了编程错误 OB121 的临时变量。

表 4-59　OB121 的变量申明表

变量	类型	描述
OB121_EV_CLASS	BYTE	事件级别和标识
OB121_SW_FLT	BYTE	故障代码
OB121_PRIORITY	BYTE	优先级=出现故障的 OB 优先级
OB121_OB_NUMBR	BYTE	OB 号



（续）

变量	类型	描述
OB121_BLK_TYPE	BYTE	出现故障块的类型（在 S7-300 PLC 时无有效值在这里记录）
OB121_RESERVED_1	BYTE	备用
OB121_FLT_REG	WORD	故障源（根据代码）。如：转换故障发生的寄存器；不正确的地址（读/写故障）；不正确的定时器/计数器/块号码；不正确的存储器区
OB121_BLK_NUM	WORD	引起故障的 MC7 命令的块号码（S7-300 PLC 无效）
OB121_PRG_ADDR	WORD	引起故障的 MC7 命令的块号码（S7-300 PLC 无效）
OB121_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

OB121 程序在 CPU 执行错误时执行，此错误不包括用户程序的逻辑错误和功能错误等，例如当 CPU 调用一个未下载到 CPU 中的程序块，CPU 会调用 OB121，通过临时变量“OB121_BLK_TYPE”可以得出出现错误的程序块。使用 STEP 7 不能实时监控程序的运行，可以用“变量表（Variable Table）”监控实时数据的变化。

打开事先已经插入的 OB121 编写程序，如图 4−134 所示。

接着在项目“Blocks”下插入 FC1，打开 FC1 编写程序，如图 4−135 所示。

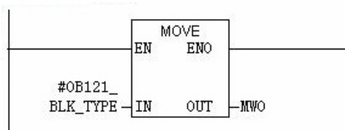


图 4−134　OB121 中编写的程序



图 4−135　FC1 中编写的程序

然后打开 OB1 编写程序，如图 4−136 所示。

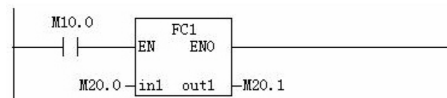


图 4−136　OB1 中编写的程序

先将硬件和 OB1 下载到 CPU 中，此时 CPU 能正常运行。在“Blocks”下插入“变量表（Variable Table）”，然后打开，填入 MW0 和 M10.0，并单击“监控”按钮，程序运行正常。将 M10.0 置为“true”后，CPU 就报错停机，查看 CPU 的诊断缓冲区信息，发现为编程错误，若将 OB121 也下载到 CPU 中，再将 M10.0 置为“true”，CPU 会报错但不停机，MW0 为“W#16#88”，“W#16#88”表示为 OB 程序错误，检查发现 FC1 未下载。下载 FC1 后，在将 M10.0 置为“true”，这时 CPU 不会再报错，程序也不会再调用 OB121。

② I/O 访问错误组织块（OB122）。当对于模块的数据访问出现故障时 CPU 的操作系统调用 OB122，OB122 与被中断的块在同一优先级中执行，表 4−60 描述了 I/O 访问错误 OB122 的临时变量。

表 4-60　OB122 的变量申明表

变量	类型	描述
OB122_EV_CLASS	BYTE	事件级别和标识
OB122_SW_FLT	BYTE	故障代码
OB122_PRIORITY	BYTE	优先级=出现故障的 OB 的优先级
OB122_OB_NUMBR	BYTE	OB 号
OB122_BLK_TYPE	BYTE	出现故障块的类型（在 S7-300 PLC 时无有效值在这里记录）
OB122_MEM_AREA	BYTE	存储器区和访问类型：位 7～4，访问类型-0、位访问-1、字节访问-2、字访问-3；位 3～0，存储器区-0、I/O 区-1、过程映像输入或输出-2
OB122_MEM_ADDR	WORD	出现故障的存储器地址
OB122_BLK_NUM	WORD	引起故障的 MC7 命令的块号码（S7-300 PLC 无效）
OB122_PRG_ADDR	WORD	引起故障的 MC7 命令的块号码（S7-300 PLC 无效）
OB122_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

7．背景组织块

CPU 可以保证设置的最小扫描循环时间，如果它比实际的扫描循环时间长，在循环程序结束后 CPU 处于空闲的时间内可以执行背景组织块（OB90）。背景 OB 的优先级为 29（最低）。OB90 中的程序是对时间要求不严格的程序。

8．启动组织块及其应用

（1）CPU 模块的启动方式

① 暖启动（Warm Restart）。S7-300 PLC CPU（不包括 CPU318）只有暖启动。过程映像数据以及非保持的 M/T/C。有保持功能的 M/T/C/DB 将保留原数值。模式开关由 STOP 扳到 RUN 位置。

② 热起动（Hot Restart 仅 S7-400 PLC 有）。在 RUN 状态时如果电源突然丢失，然后又重新上电，从上次 RUN 模式结束时程序被中断之处继续执行，不对计数器等复位。

③ 冷启动（Cold Restart，CPU 417 和 CPU 417H）。冷启动时，过程数据区的 I、Q、M、T、C、DB 等被复位为零。模式开关扳到 MRES 位置。

（2）启动组织块（OB100～OB102）

在暖启动、热启动或冷起动时，操作系统分别调用 OB100、OB101 或 OB102。

【例 4-22】 编写一段初始化程序，将 CPU 314-2DP 的 MB0～MB3 单元清零。

【解】

一般初始化程序在 CPU 一启动后就运行，CPU 314-2DP 只有热启动方式，所以只能使用 OB100 组织块。MB0～MB3 实际上就是 MD0，其程序如图 4-137 所示。

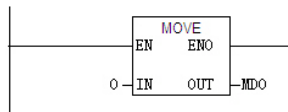


图 4-137　OB100 中编写的程序



4.6.3　STL源文件

编程语言 STL 是一种基于文本的编程语言，它具有一种类似机器代码的结构。每个语句代表 CPU 的一种程序处理操作。多个语句连接在一起就构成了程序段。编程语言 STL 随标准 STEP 7 软件包提供。通过该编程语言，可使用增量编辑器编辑 S7 块，并可在源代码编辑器中创建和编译 STL 程序源文件以生成块。

1．创建STL源文件，并转化成块程序

用一个例子说明创建 STL 源文件，并转化成块程序的过程。

【例 4-23】 创建电动机起停控制的 STL 源文件，并转化成块程序。

【解】

首先创建一个工程，工程名称为“STL 源文件（1）”，先选中“源文件”，在菜单中单击“插入”→“S7 软件”→“S7 源文件”，如图 4-138 所示。



图 4-138　插入源文件（1）

如图 4-139 所示，双击“STL 源文件（1）”，打开 STL 源文件（1）的编辑器。在菜单中单击“插入”→“块模板”→“FC”，插入源文件 FC 模板，如图 4-140 所示。

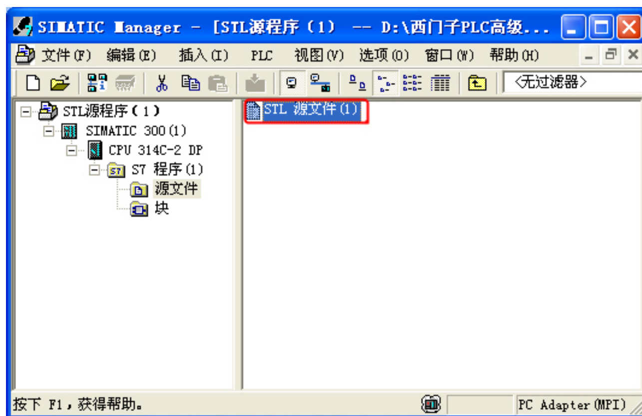


图 4-139　插入源文件（2）



图 4-140　插入源文件 FC 模板

如图 4-141 所示编辑器中，源文件已经具备基本框架，先在“1”处编写程序，其余的地方保持不变，再单击工具栏的“编译”按钮，编译结果为“0 个错误，0 个警告”，保存源文件。

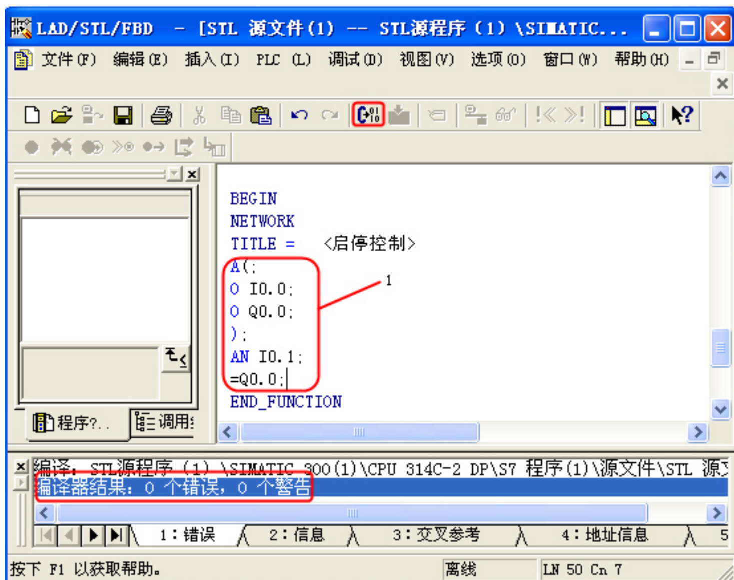


图 4-141　编写程序

回到管理器界面，如图 4-142 所示，选中“块”，双击“FC1”，即可打开功能 FC1，只要简单切换，就可变成如图 4-143 所示的梯形图。

2．利用STL源文件，实现程序保护

若读者编写的程序需要进行知识产权保护，那么可以用源文件实现这个目的。以下用一个例子说明利用 STL 源文件实现程序保护的过程。





图 4-142　打开 FC1

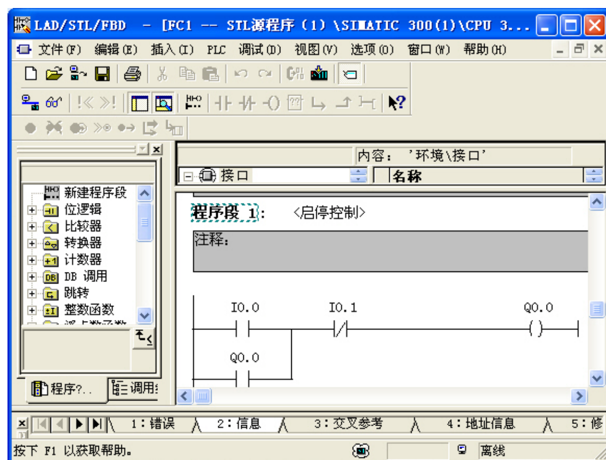


图 4-143　FC1 的梯形图

【例 4-24】　创建电动机起停控制的 STL 源文件，并保护此源程序。

【解】

先插入一个功能 FC2，如图 4-144 所示，双击“FC2”打开功能的程序编辑器界面。

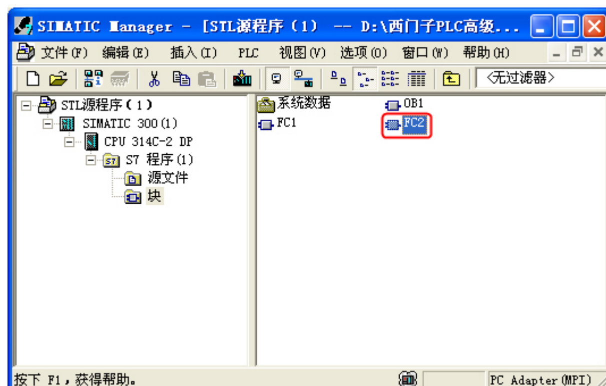


图 4-144　打开 FC2

在 FC2 的程序编辑器里，新建变量“Start”、“Halt”和“Motor”，再编写梯形图程序如图 4-145 所示。

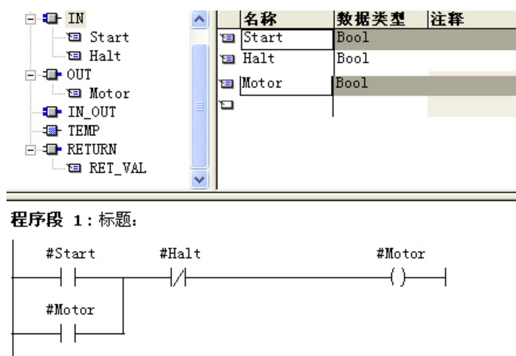


图 4-145　FC2 中的程序

把 FC2 的块转化成 STL 源文件。如图 4-146 所示，单击菜单栏的“文件”→“生成源文件”，自动弹出“新建源文件”界面，如图 4-147 所示，将对象名称命名为“STL 源文件（2）”，单击“确定”按钮。

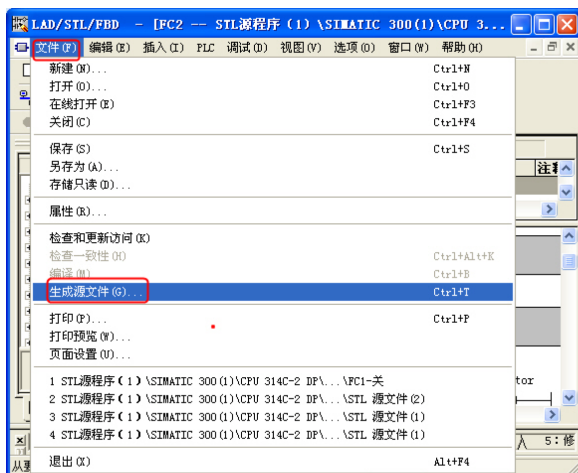


图 4-146　生成源文件

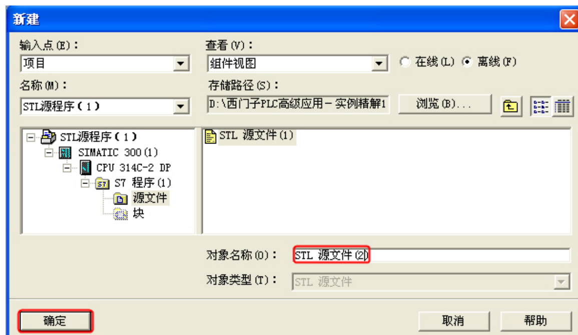
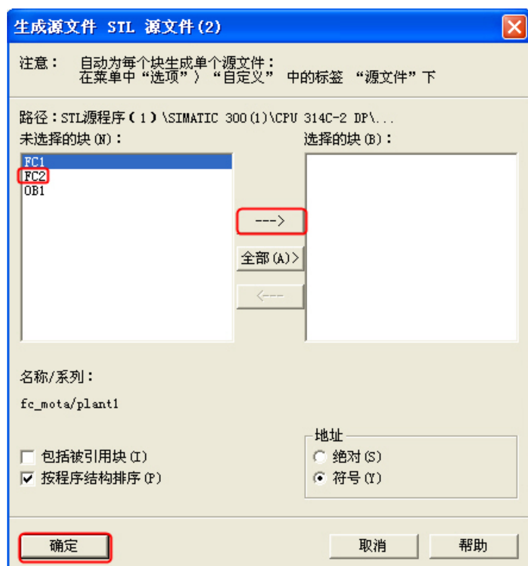


图 4-147　新建源文件



先选中“FC2”，再单击“选中”按钮 --->，可以看到左侧的“未选择的块”框中，移到右侧“选择的块”框中，最后单击“确定”按钮，如图 4-148 所示，接着弹出确认对话框，如图 4-149 所示，单击“确定”按钮。



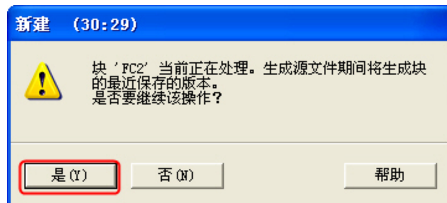


图 4-148 选择要转换的块　　　　图 4-149 确认对话框

返回到管理器界面，选中“源文件”，再双击“STL 源文件（2）”，打开源文件，如图 4-150 所示。

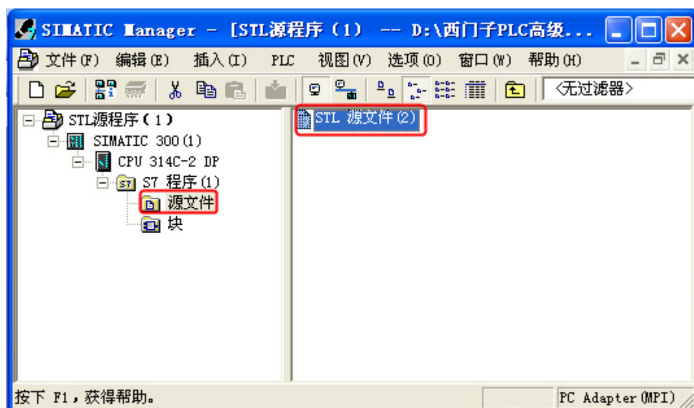


图 4-150 打开源文件

在如图 4-151 所示界面的“1”处，加入关键字“KNOW_HOW_PROTECT”，再在工具栏上单击“编译”按钮，如果编译成功，那么 FC2 将会被保护，回到管理器界面，可以看到 FC2 已经加锁，如图 4-152 所示，此时双击“FC2”，是不能看到以前写在编译器中的程序。

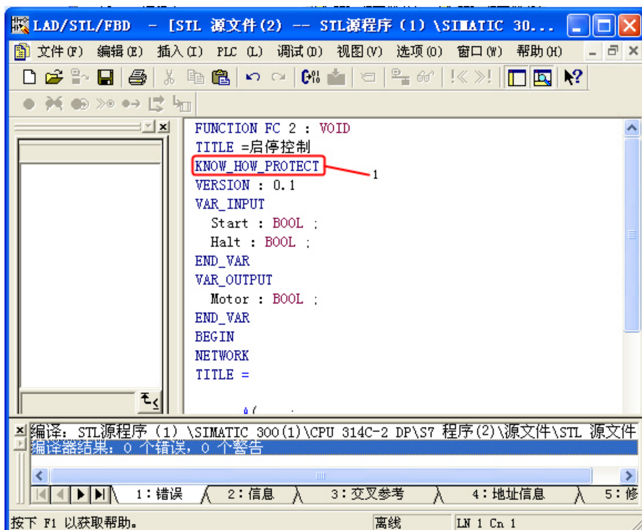


图 4-151　加关键字“KNOW_HOW_PROTECT”

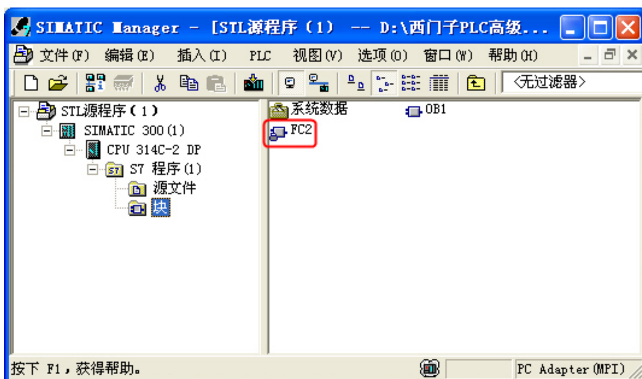


图 4-152　管理界面中被保护的块

4.7　实例

至此，读者已经对 S7-300/400 PLC 的软硬件已经有一定的了解，本节内容将列举一些简单的例子，供读者模仿学习。

【例 4-25】 请编写三相异步电动机的 Y-△（星-三角）起动控制程序。

【解】

为了让读者对用 S7-300/400 PLC 作为控制器的工程有一个完整的了解，本例比较详细地描述整个控制过程。

（1）软硬件的配置

① 1 套 STEP 7 V5.4 SP4。

② 1 台 CPU 314C-2DP。

③ 1 根编程电缆（或者 CP5611 卡）。



④ 电动机、接触器和继电器等。

电动机 Y-△减压起动主回路图和接线图如图 4-153 和图 4-154 所示。CPU 314C-2DP 是经济型模块，只有晶体管输出形式，PLC 要控制接触器，必须加一级中间继电器。

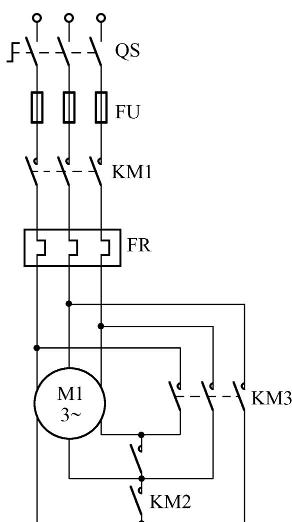


图 4-153 电动机 Y-△减压起动主回路图

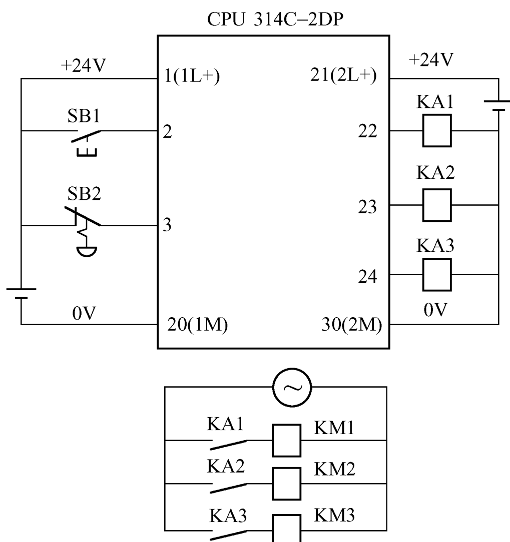


图 4-154 电动机 Y-△减压起动 PLC 接线图

【关键点】输出模块（如 SM322，订货号 6ES7 322-1HH01-0AA0）是继电器输出形式，可以直接驱动接触器，但是笔者仍然推荐加一级中间继电器，理由很明显，输出模块中的继电器的驱动能力一般较小，损坏后不易更换，而外置的中间继电器的驱动能力是可以选择的，而且损坏后很容易更换。

此外，急停按钮一般使用常闭触点，若使用常开触点，单从逻辑上是可行的，但在某些极端情况下，当接线意外断开时，急停按钮是不能起停机作用的，容易发生事故。这一点请读者务必注意。

（2）硬件组态

1）新建工程。打开 STEP 7，新建工程，命名为“星三角起动”，如图 4-155 所示。

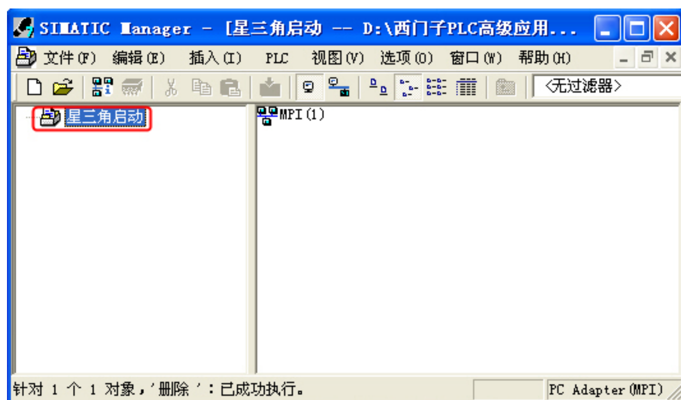


图 4-155 新建工程

2）插入站点。选中“星三角起动”，在菜单栏中，单击“插入”→“站点”→“SIMATIC 300 站点”，如图 4-156 所示。



图 4-156　插入站点

3）打开程序硬件组态。双击“硬件”，如图 4-157 所示，之后弹出硬件组态界面。

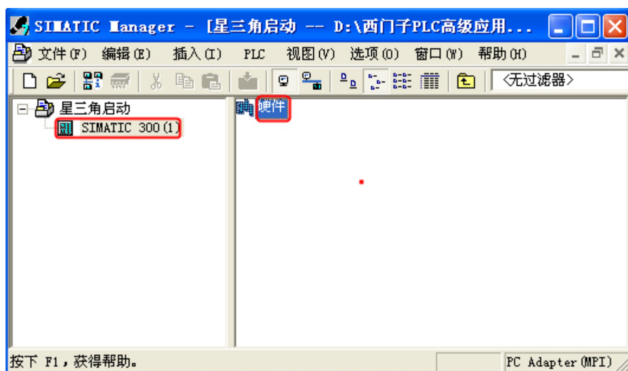


图 4-157　打开硬件组态

4）添加导轨。在硬件组态界面，先选中导轨“Rail”，按住鼠标的左键不放，将导轨拖入左侧的空白处，如图 4-158 所示，松开鼠标左键即可。

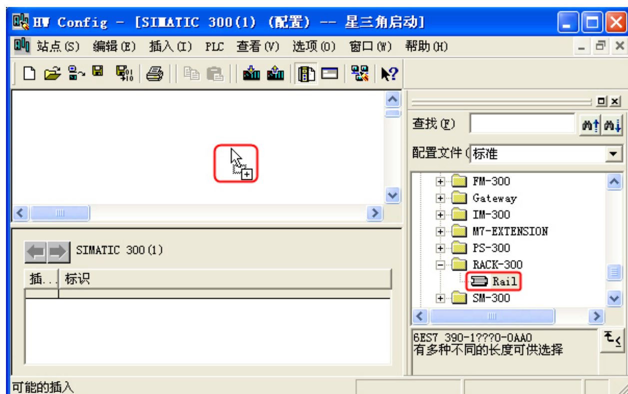


图 4-158　添加导轨



5）插入 CPU 模块。选中“CPU 314C”的“V2.6”，按下鼠标的左键不放，将选中的 CPU 拖到导轨的槽位 2，如图 4-159 所示。

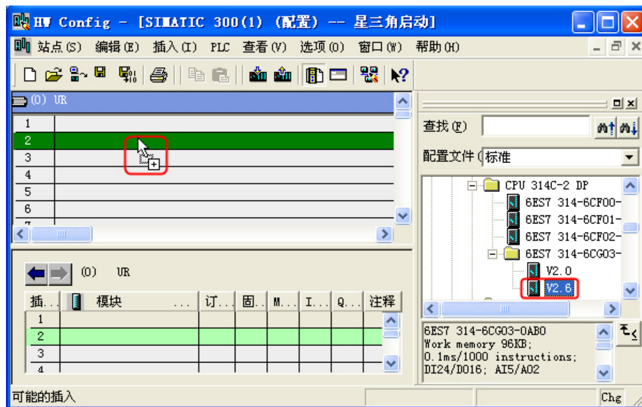


图 4-159 管理界面中被保护的块

6）打开输入/输出属性。如图 4-160 所示，“2”处的输入地址是 IB124～IB126，输出地址是 QB124～QB125，这不符合习惯，可以修改输入/输出属性，选中“DI24/DO16”，并双击之，便可打开输入/输出属性。

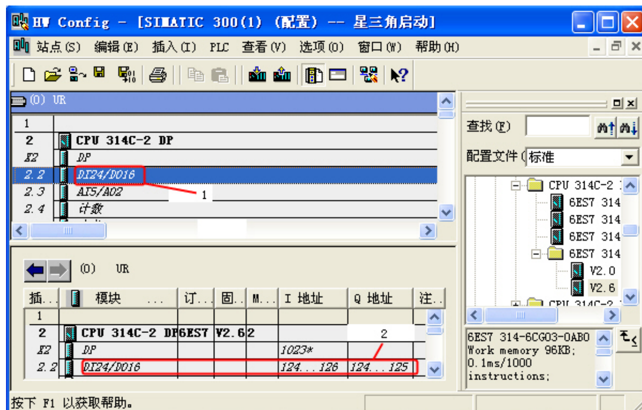


图 4-160 打开输入/输出属性

7）更改输入/输出的起始地址。先去掉“系统默认”前面的“√”，再在“开始”后面的方框中输入起始地址 0，最后单击“确定”按钮即可，如图 4-161 所示。回到硬件组态界面如图 4-160 所示，单击工具栏中的“编译和保存”按钮，对硬件组态进行编译和保存，若有错误，会弹出警告。

（3）编写程序、调试程序

1）打开程序编辑器。回到管理器界面，选中“块”，双击“OB1”，弹出程序编辑器界面，如图 4-162 所示。

2）编写程序。在程序编辑器编写如图 4-163 所示的程序。首先闭合电源开关，接通总电源，再接通起动开关（I0.0），使电动机绕组实现Y形连接（Q0.1），延时 5s 后，电动机绕组改为△形连接（Q0.2）。按下停止按钮（I0.1），电动机停转。







图 4-161　更改输入/输出的起始地址

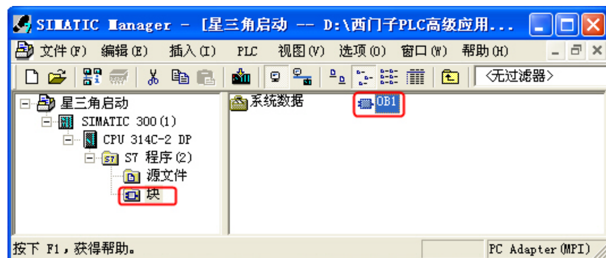


图 4-162　打开程序编辑器

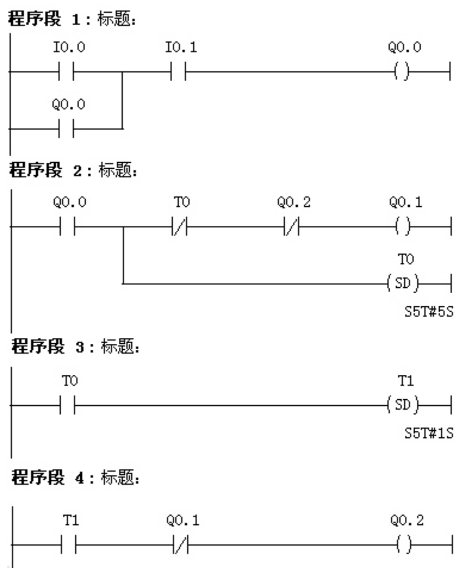


图 4-163　打开程序编辑器



【例 4-26】 十字路口的交通灯控制，当合上起动按钮时，东西方向的绿灯亮 4s，闪烁 2s 后灭；黄灯亮 2s 后灭；红灯亮 8s 后灭，如此循环；而对应东西方向绿灯、红灯、黄灯亮时，南北方向红灯亮 8s 后灭；接着绿灯亮 4s，闪烁 2s 后灭；黄灯亮 2s 后灭，如此循环。请画出接线图，并编写 PLC 控制程序。

【解】

（1）软硬件的配置

① 1 套 STEP 7 V5.4 SP4。

② 1 台 CPU 314C-2DP。

③ 1 根编程电缆（或者 CP5611 卡）。

④ 电缆和继电器等。

（2）硬件组态

硬件组态的常规步骤参考例 4-25，在此不做赘述，题目中要求绿灯闪烁，这可以在硬件组态时设置。如图 4-164 所示，在硬件组态界面中，选中“CPU 314C-2DP”，并双击之，弹出 CPU 的属性界面，选中“周期/时钟存储器”选项卡，勾选“时钟存储器”，并在它后面的方框中输入 100，如图 4-165 所示。这样操作的目的是将 MB100 确定为时钟存储器，其中 M100.5 的脉冲频率为 1Hz，具体可参考前面的章节。

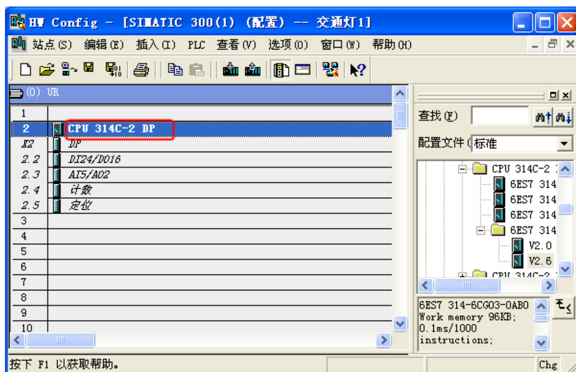


图 4-164 打开 CPU 的属性

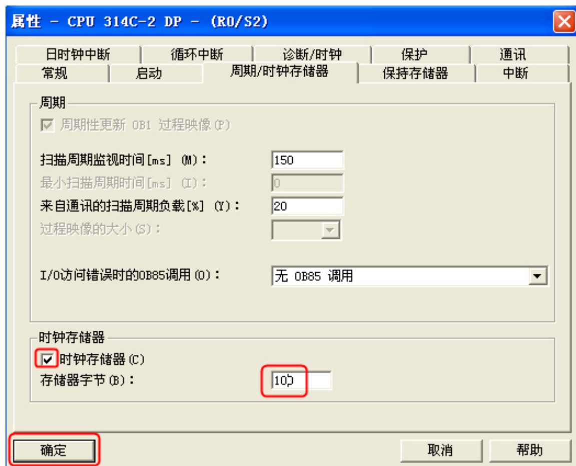


图 4-165 确定时钟存储器为 MB100

【关键点】修改时钟存储器的组态信息必须编译和保存，再下载到 CPU 中，才起作用，许多初学者，经常忘记把修改后的硬件组态或者修改后的程序下载到 CPU 中，这是要避免的。

（3）I/O 分配和接线

首先根据题意画出东西和南北方向 3 种颜色灯的亮灭序图，再进行 I/O 分配。

输入：启动-I0.0；停止-I0.1。

输出（东西方向）：红灯-Q1.0，黄灯-Q1.1；绿灯-Q1.2。

输出（南北方向）：红灯-Q0.0，黄灯-Q0.1；绿灯-Q0.2。

东西和南北方向各有 3 盏，从时序图容易看出，共有 6 个连续的时间段，因此要用到 6 个定时器，这是解题的关键。用这 6 个定时器控制两个方向 6 盏灯的亮或灭，不难设计梯形图。交通灯时序图、I/O 接线图和交通灯梯形图分别如图 4-166～图 4-169 所示。

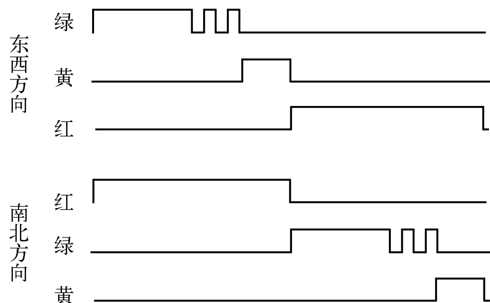


图 4-166　交通灯时序图

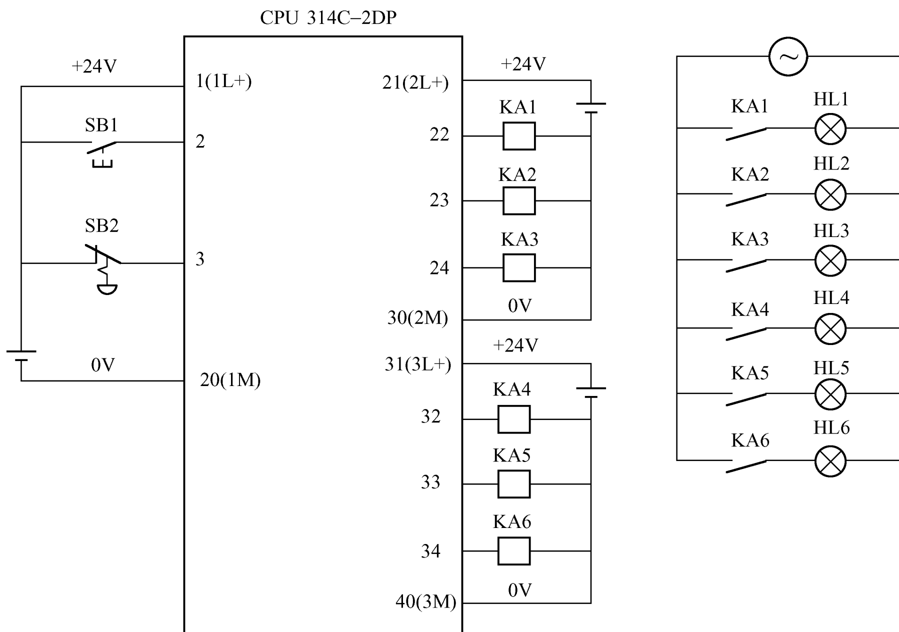
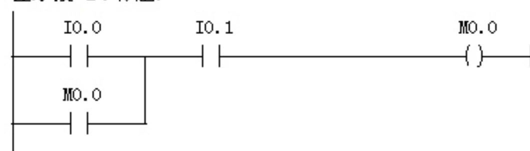
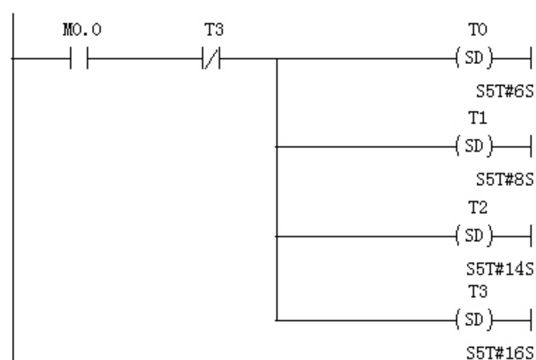
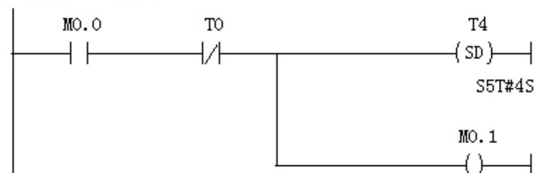


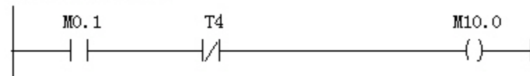
图 4-167　接线图

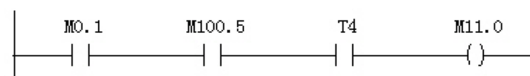














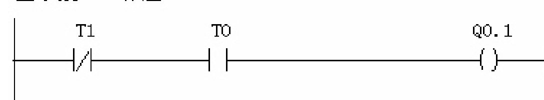
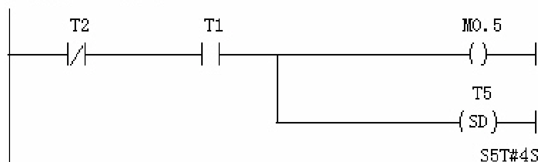
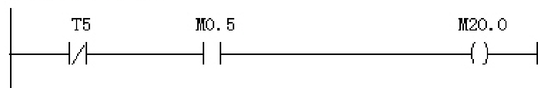
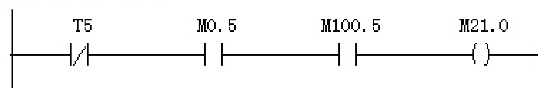


图 4-168　交通灯梯形图（基本指令）

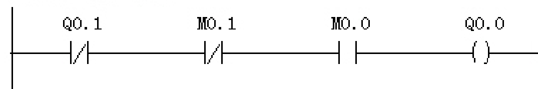












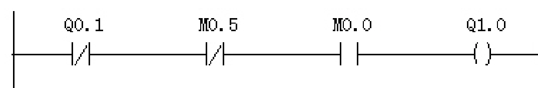


图 4-168　交通灯梯形图（基本指令）（续）

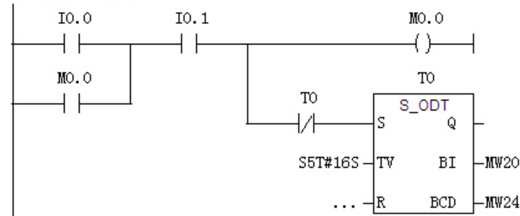
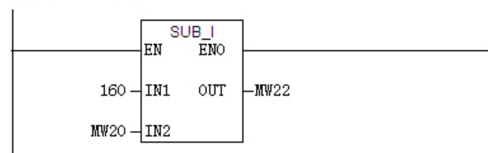


图 4-169　交通灯梯形图（比较指令）



程序段 2：标题：



程序段 3：标题：

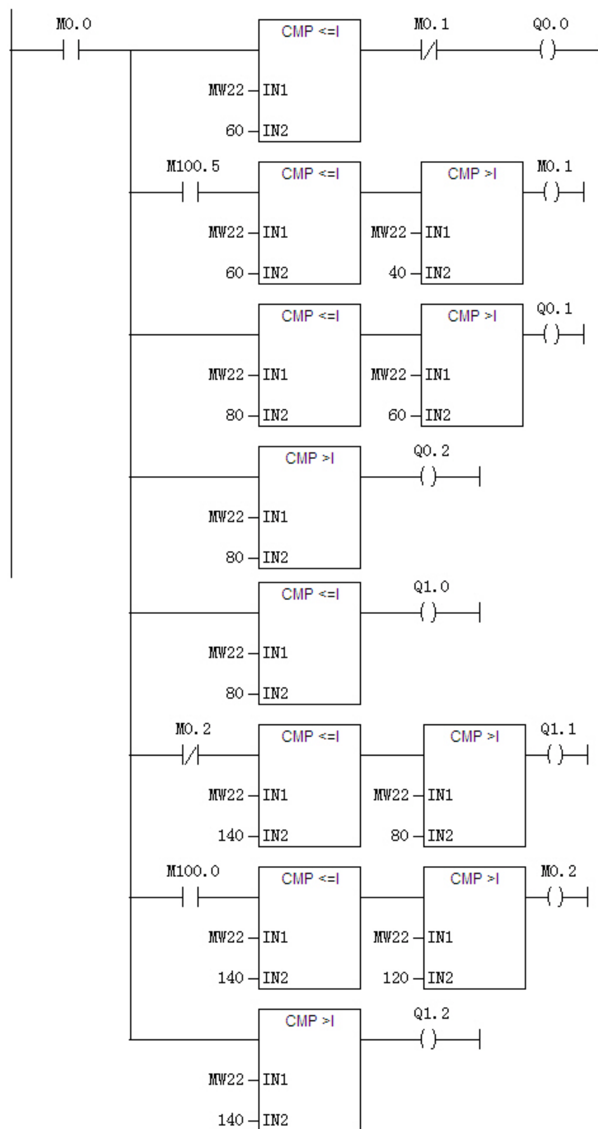


图 4-169　交通灯梯形图（比较指令）（续）

用基本指令编写交通灯的程序用到 6 个定时器，程序也比较长，阅读程序比较费力，而用比较指令编写程序，程序中只用 1 个定时器，程序的长度比较短，也比较容易阅读，程序如图 4-170 所示。程序 MW20 中的数值剩余的时间，其单位是 0.1s，因此 MW22=160–

MW20，表示 MW22 中的数值是当前时间，其单位为 0.1s。

【例 4-27】 有 3 台电动机，由 Q0.1～Q0.3 的输出点控制，使用单按钮控制起/停。按钮接在 I0.0 上，具体的控制方法是：按下按钮的次数对应起动电动机的号码，按第 4 次时所有的电动机停止，任何时候按下按钮持续 3s 时，电动机停止。

【解】梯形图如图 4-170 所示。

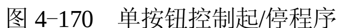


图 4-170　单按钮控制起/停程序

【例 4-28】 有一个控制系统，控制器是 CPU 314C-2DP，压力传感器测量油压力，油压力的范围是 0～40MPa，当油压力高于 30MPa 时报警，请设计此系统。

【解】

CPU 314C-2DP 集成了有模拟量输入/输出和数字量输入/输出，其接线如图 4-171 所示，模拟量输入的端子 2 和 4 分别与传感器的电压信号+和电压信号－相连，传感器的电源在图中未表示。

数值转换（FC105）SCALE 功能接受一个整型值（IN），并将其转换为以工程单位表示的介于下限和上限（LO_LIM 和 HI_LIM）之间的实型值。将结果写入 OUT。SCALE 功能使用以下等式：



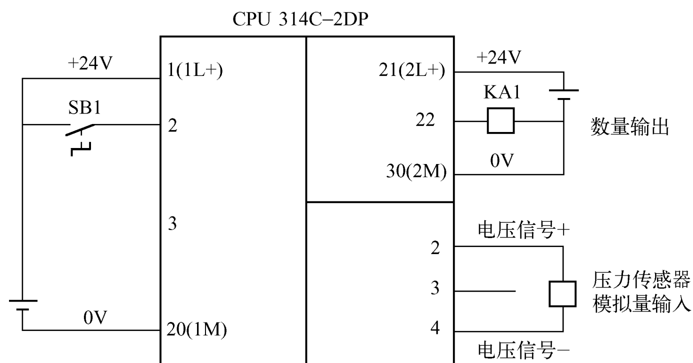


图 4–171 接线图

$$OUT = \frac{IN - K1}{K2 - K1}(HI_LIM - LO_LIM) + LO_LIM$$

常数 K1 和 K2 根据输入值是 BIPOLAR（双极性）还是 UNIPOLAR（单极性）设置。为 BIPOLAR 时，假定输入整型值介于–27648 与 27648 之间，因此 K1=–27648.0，K2=+27648.0。为 UNIPOLAR 时，假定输入整型值介于 0 和 27648 之间，因此 K1=0.0，K2= +27648.0。

本例是单极性，故 K1=0.0，K2=+27648.0。上极限 HI_LIM 为 40 和下极限 LO_LIM 为 0。模拟量采集到 PIW752 中，这个数值的范围是 0～27648，经过数值转化后，MW14（OUT）中的数值为 0～40，是油压力。当然这个题目也可以不用转换函数 FC105，直接用数学函数，但要麻烦得多。

梯形图如图 4–172 和图 4–173 所示。

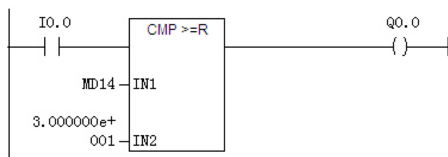


图 4–172 OB1 中的程序

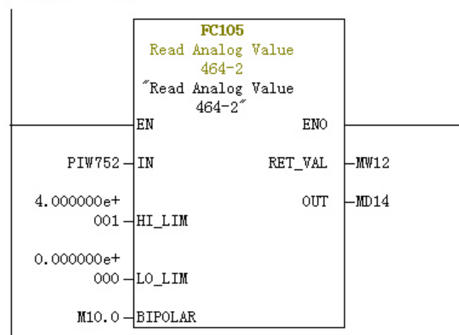


图 4–173 OB35 中的程序

小结

重点难点总结

本章的内容比较多，重难点内容多，读者必须掌握的内容有以下几点：

1．数据类型。这是学习任何 PLC 都必须掌握的内容，S7-300/400 PLC 的数据类型多，

应用灵活。

2．指令。S7-300/400 PLC 的指令多，但大部分指令还是比较容易学会的，对照 STEP 的帮助，一般的读者可以学会。定时器指令、移位指令和装载指令都较难，必须掌握。

3．功能、功能块、系统功能、系统功能块和数据块的应用很灵活，也是难点，需要读者掌握。

4．组织和中断。计算机系统都有中断，中断对任何计算机系统都是难点和重点，S7-300/400 PLC 的中断组织多、比较复杂，是学习重点和难点。

习题

1．用 PLC 设计一个闹钟，每天早上 6:00 闹铃。

2．用 PLC 的置位、复位指令实现彩灯的自动控制。控制过程为：按下起动按钮，第一组花样绿灯亮；10s 后第二组花样蓝灯亮；20s 后第三组花样红灯亮，30s 后返回第一组花样绿灯亮，如此循环，并且仅在第三组花样红灯亮后方可停止循环。

3．如图 4−174 所示为一台电动机起动的工作时序图，试画出梯形图。

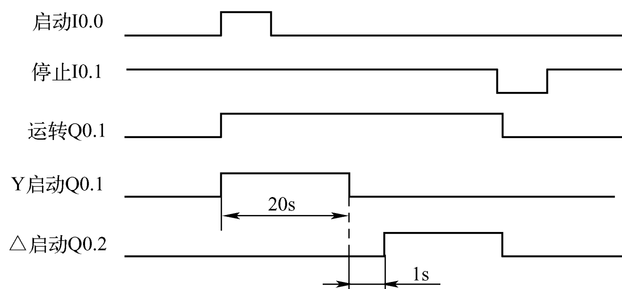


图 4−174　习题 3 附图

4．用 3 个开关（I0.1、I0.2、I0.3）控制一盏灯 Q1.0，当 3 个开关全通或者全断时灯亮，其他情况灯灭。（提示：使用比较指令。）

5．用移位指令构成移位寄存器，实现广告牌字的闪耀控制。用 HL1～HL4，4 只灯分别照亮“欢迎光临”4 个字，其控制要求见表 4−61，每步间隔 1s。

表 4−61　广告牌字闪耀流程

流程	1	2	3	4	5	6	7	8
HL1	√				√		√	
HL2		√			√		√	
HL3			√		√		√	
HL4				√	√		√	

6．运用算术运算指令完成算式[（100+200）×10]/3 的运算，并画出梯形图。

7．编写一段程序，将 MB100 开始的 50 个字的数据传送到 MB1000 开始的存储区。

8．现有 3 台电动机M_1、M_2、M_3，要求按下起动按钮I0.0 后，电动机按顺序起动（M_1



起动，接着M_2起动，最后M_3起动），按下停止按钮I0.1 后，电动机按顺序停止（M_3先停止，接着M_2停止，最后M_1停止）。试设计其梯形图并写出指令表。

9．如图 4-175 所示，若传送带上 20s 内无产品通过则报警，并接通 Q0.0。试画出梯形图并写出指令表。

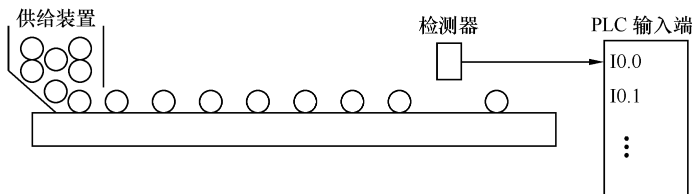


图 4-175　习题 9 附图

10．如图 4-176 所示为两组带机组成的原料运输自动化系统，该自动化系统的起动顺序为：盛料斗 D 中无料，先起动带机 C，5s 后再起动带机 B，经过 7s 后再打开电磁阀 YV，该自动化系统停机的顺序恰好与起动顺序相反。试完成梯形图设计。

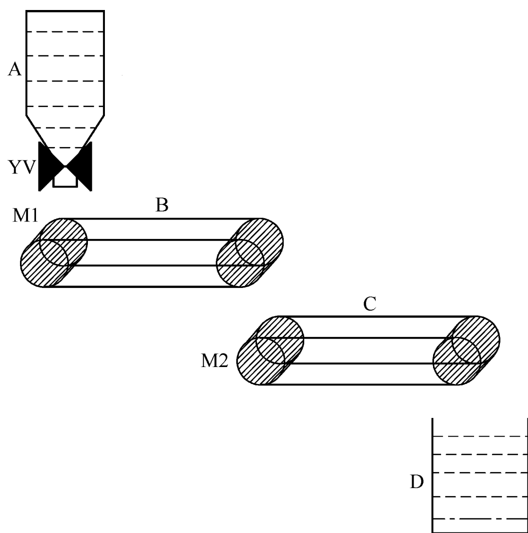


图 4-176　习题 10 附图

11．编写一段检测上升沿变化的程序。每当 I0.1 接通一次，MB0 的数值增加 1，如果计数达到 18 时，Q0.1 接通，用 I0.2 使 Q0.1 复位。

12．设计一个照明电路，当按下 I0.0 按钮，照明灯 Q0.0 亮 10s，如果在这段时间里有人按下按钮，时间从头开始，确保最后一次按下按钮灯能亮 30s。

13．自动生产线上，要统计出每天产品的生产量，当产品经过传感器感应区时，PLC 计数一次，若一天以 8 小时计算，要求编制一天产量的程序，并设计接线图。

第 5 章

S7-300/400 PLC的编程方法与调试

本章介绍功能图的画法、梯形图的禁忌以及如何根据功能图用基本指令、功能指令和复位置位指令 3 种方法编写顺序控制梯形图。另一个重要的内容是程序的调试方法。

5.1　功能图

5.1.1　功能图的画法

功能图（SFC）是描述控制系统的控制过程、功能和特征的一种图解表示方法。它具有简单、直观等特点，不涉及控制功能的具体技术，是一种通用的语言，是 IEC（国际电工委员会）首选的编程语言，近年来在 PLC 的编程中已经得到了普及与推广。

功能图的基本思想是：设计者按照生产要求，将被控设备的一个工作周期划分成若干个工作阶段（简称“步”），并明确表示每一步要执行的输出，“步”与“步”之间通过制定的条件进行转换，在程序中，只要通过正确连接进行“步”与“步”之间的转换，就可以完成被控设备的全部动作。

PLC 执行功能图程序的基本过程是：根据转换条件选择工作“步”，进行“步”的逻辑处理。组成功能图程序的基本要素是步、转换条件和有向连线，如图 5–1 所示。

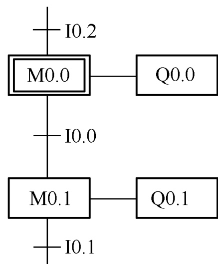


图 5–1　功能图

1．步

一个顺序控制过程可分为若干个阶段，也称为步或状态。系统初始状态对应的步称为初始步，初始步一般用双线框表示。在每一步中施控系统要发出某些“命令”，而被控系统要完成某些“动作”、“命令”和“动作”都称为动作。当系统处于某一工作阶段时，则该步处于激活状态，称为活动步。

2．转换条件

使系统由当前步进入下一步的信号称为转换条件。顺序控制设计法用转换条件控制代表各步的编程元件，让它们的状态按一定的顺序变化，然后用代表各步的编程元件去控制输出。不同状态的“转换条件”可以不同，也可以相同。当“转换条件”各不相同时，在功能图程序中每次只能选择其中一种工作状态（称为“选择分支”），当“转换条件”都相同时，在功能图程序中每次可以选择多个工作状态（称为“选择并行分支”）。只有满足条件状态，才能进行逻辑处理与输出。因此，“转换条件”是功能图程序选择工作状态（步）的“开关”。



3．有向连线

步与步之间的连接线称为“有向连线”，“有向连线”决定了状态的转换方向与转换途径。在有向连线上有短线，表示转换条件。当条件满足时，转换得以实现，即上一步的动作结束而下一步的动作开始，因而不会出现动作重叠。步与步之间必须要有转换条件。

图 5-1 中的双框为初始步，M0.0 和 M0.1 是步名，I0.0、I0.1 为转换条件，Q0.0、Q0.1 为动作。当 M0.0 有效时，输出指令驱动 Q0.0。步与步之间的连线称为有向连线箭头省略未画。

4．功能图的结构分类

根据步与步之间的进展情况，功能图分为以下 3 种结构。

（1）单一顺序

单一顺序动作是一个接一个地完成，完成每步只连接一个转移，每个转移只连接一个步，如图 5-2a 所示。根据功能图很容易写出代数逻辑表达式，代数逻辑表达式和梯形图有对应关系，由代数逻辑表达式可写出梯形图，如图 5-2b 所示。图 5-2c 和图 5-2b 的逻辑是等价的，但图 5-2c 更加简洁（程序的容量要小一些），因此经过 3 次转化，最终的梯形图是图 5-2c。

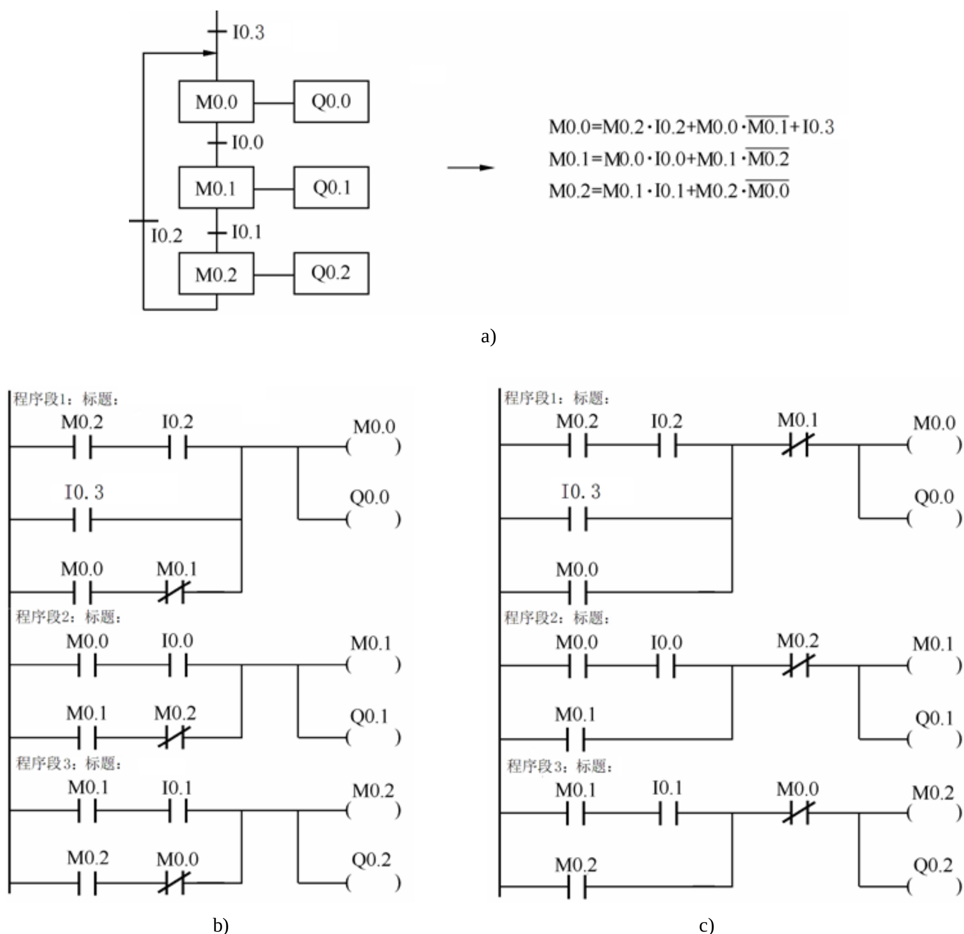


图 5-2　单一顺序

（2）选择顺序

选择顺序是指某一步后有若干个单一顺序等待选择，称为分支，一般只允许选择进入一个顺序，转换条件只能标在水平线之下。选择顺序的结束称为合并，用一条水平线表示，水平线以下不允许有转换条件，如图 5-3 所示。

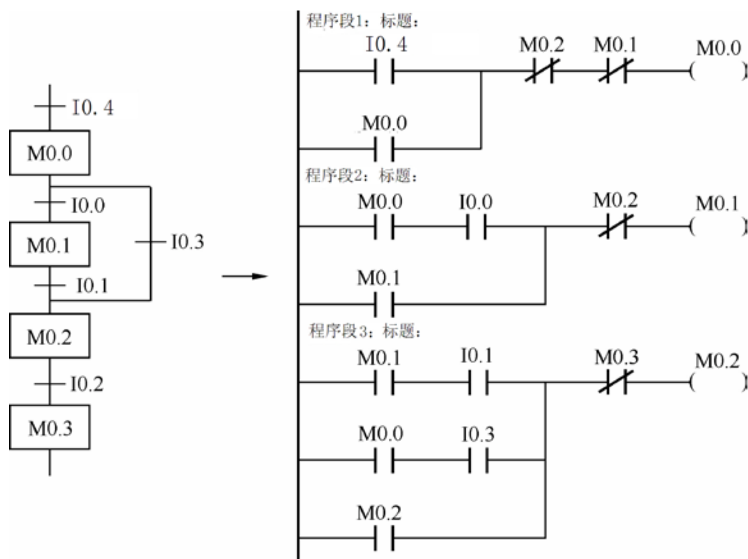


图 5-3　选择顺序

（3）并行顺序

并行顺序是指在某一转换条件下同时启动若干个顺序，也就是说转换条件实现导致几个分支同时激活。并行顺序的开始和结束都用双水平线表示，如图 5-4 所示。

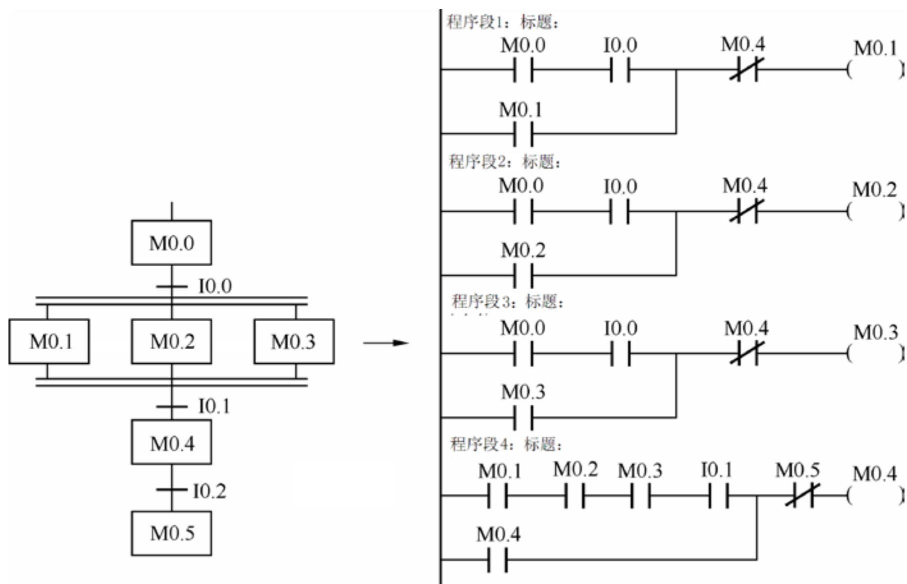


图 5-4　并行顺序



【例 5–1】 某系统的功能图如图 5–5 所示，请设计对应的梯形图。

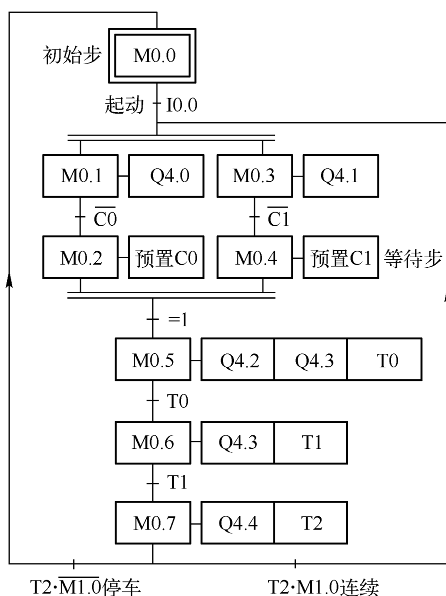


图 5–5 功能图

【解】

对应的梯形图如图 5–6 所示。

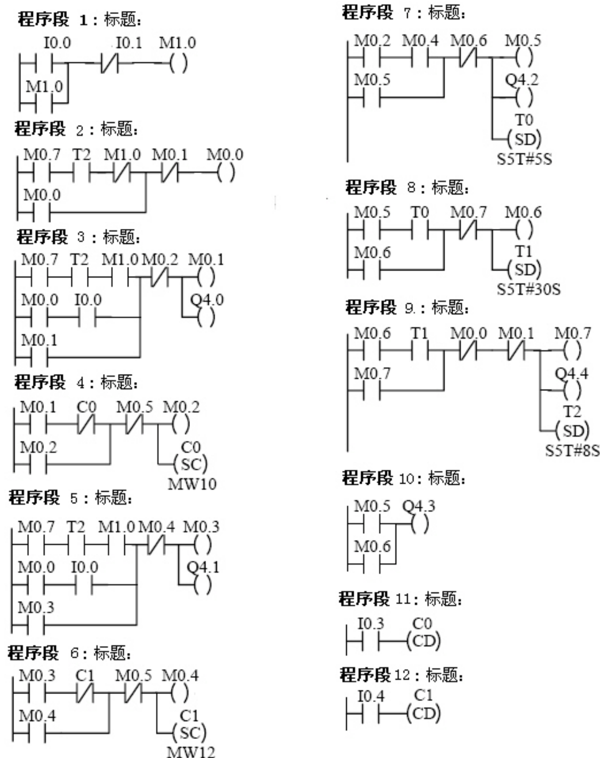


图 5–6 梯形图

5．功能图设计的注意点

1）状态之间要有转换条件。如图 5−7 所示，状态之间缺少“转换条件”是不正确的，应改成如图 5−8 所示的功能图。必要时转换条件可以简化，如将图 5−9 简化成图 5−10。

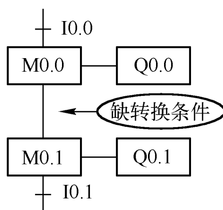


图 5−7　错误的功能图

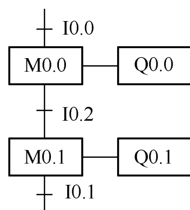


图 5−8　正确的功能图

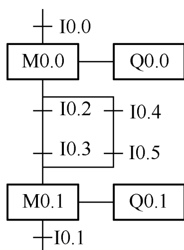


图 5−9　简化前的功能图

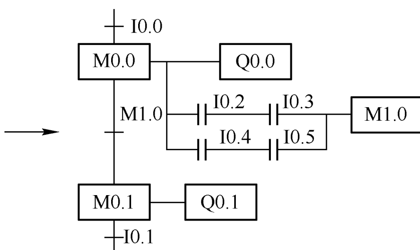


图 5−10　简化后的功能图

2）转换条件之间不能有分支。例如，图 5−11 应该改成如图 5−12 所示的合并后的功能图，合并转换条件。

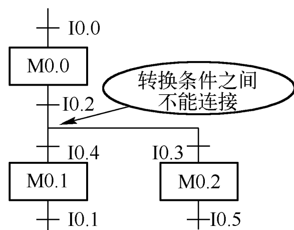


图 5−11　错误的功能图

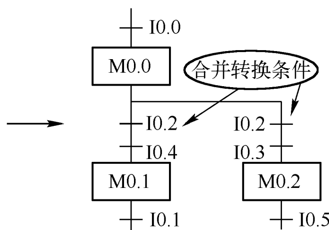


图 5−12　合并后的功能图

3）顺序功能图中的初始步对应于系统等待启动的初始状态，初始步是必不可少的。

4）顺序功能图中一般应有由步和有向连线组成的闭环。

5.1.2　梯形图编程的原则

尽管梯形图与继电器电路图在结构形式、元件符号及逻辑控制功能等方面相类似，但它们又有许多不同之处，梯形图有自己的编程规则。

1）每一逻辑行总是起于左母线，最后终止于线圈或右母线（右母线可以不画出），如图 5−13 所示。

2）无论选用哪种机型的 PLC，所用元件的编号必须在该机型的有效范围内。例如 S7-



300 PLC 中没有 M99000.0。

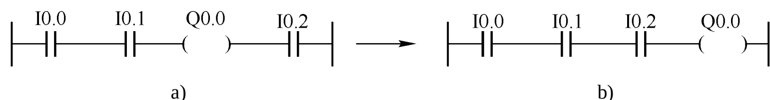


图 5-13　梯形图

a) 错误　b) 正确

3）梯形图中的触点可以任意串联或并联，但继电器线圈只能并联而不能串联。

4）触点的使用次数不受限制。例如，辅助继电器 M0.0 可以在梯形图中出现无限制的次数，而实物继电器的触点一般少于 8 对，只能用有限次。

5）在梯形图中同一线圈只能出现一次。如果在程序中，同一线圈使用了两次或多次，称为“双线圈输出”。对于“双线圈输出”，有些 PLC 将其视为语法错误，绝对不允许（如三菱 FX 系列 PLC）；有些 PLC 则将前面的输出视为无效，只有最后一次输出有效（如西门子 PLC）；而有些 PLC 在含有跳转指令或步进指令的梯形图中允许双线圈输出。

6）西门子 PLC 的梯形图中不能出现 I 线圈。

7）对于不可编程的梯形图必须经过等效变换，变成可编程梯形图，如图 5-14 所示。

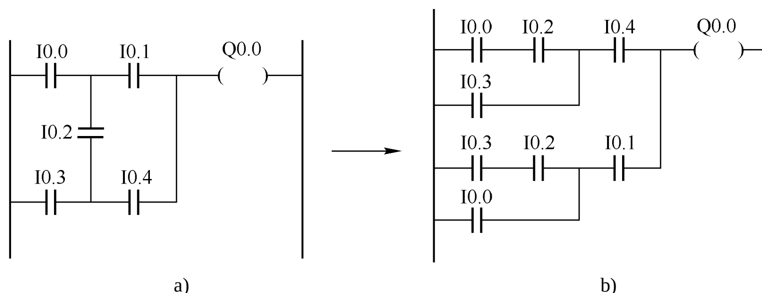


图 5-14　梯形图

a) 错误　b) 正确

8）在有几个串联电路相并联时，应将串联触点多的回路放在上方，归纳为“多上少下”的原则，如图 5-15 所示。在有几个并联电路相串联时，应将并联触点多的回路放在左方，归纳为“多左少右”原则，如图 5-16 所示。因为这样所编制的程序简洁明了，语句较少。但要注意图 5-15a 和图 5-16a 的梯形图逻辑上是正确的。

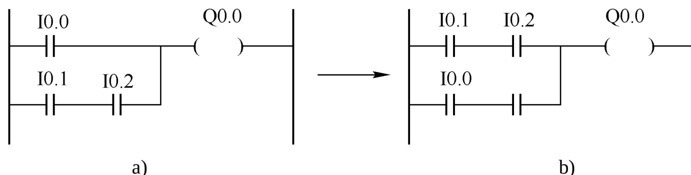


图 5-15　梯形图

a) 不合理　b) 合理

9）PLC 的输入端所连的电器元件通常使用常开触点，即使与 PLC 对应的继电器-接触器系统原来使用的是常闭触点，改为 PLC 控制时也应转换为常开触点。如图 5-17 所示为继





电器-接触器系统控制的电动机的起/停控制，如图 5-18 所示为电动机的起/停控制的梯形图，如图 5-19 所示为电动机起/停控制的接线图。可以看出：继电器-接触器系统原来使用常闭触点 SB1 和 FR，改用 PLC 控制时，则在 PLC 的输入端变成了常开触点。

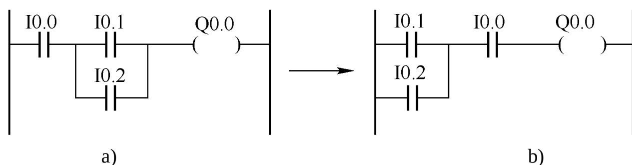


图 5-16　梯形图

a) 不合理　b) 合理

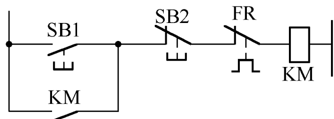


图 5-17　电动机起/停控制图

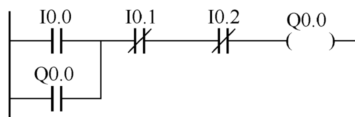


图 5-18　电动机起/停控制的梯形图

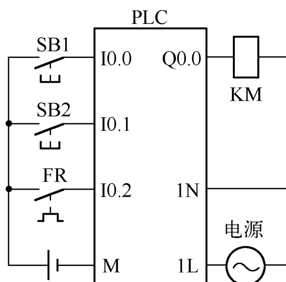


图 5-19　电动机的起/停控制的接线图

【关键点】图 5-18 的梯形图中 I0.1 和 I0.2 用常闭触点，否则控制逻辑不正确。若读者一定要让 PLC 的输入端的按钮为常闭触点输入也可以（一般不推荐这样使用），但梯形图中 I0.1 和 I0.2 要用常开触点，对于急停按钮必须使用常闭触头，若一定要使用常开触头，从逻辑上讲是可行的，但在某些情况下，有可能急停按钮不起作用而造成事故，这是读者要特别注意的。另外，一般不推荐将热继电器的常开触点接在 PLC 的输入端，因为这样做占用了宝贵的输入点，最好将热继电器的常闭触点接在 PLC 的输出端，与 KM 的线圈串联。

5.2　逻辑控制的梯形图编程方法

相同的硬件系统，由不同的人设计，可能设计出不同的程序，有的人设计的程序简洁、而且可靠，而有的人设计的程序虽然能完成任务，但较复杂，PLC 程序设计是有规律可循的，下面将介绍两种方法：经验设计法和功能图设计法。

5.2.1　经验设计法

经验设计法就是在一些典型的梯形图的基础上，根据具体的对象对控制系统的具体要

求，对原有的梯形图进行修改和完善。这种方法适合有一定工作经验的人，这些人有现成的资料，特别在产品更新换代时，使用这种方法比较节省时间。下面举例说明这种方法的思路。

【例 5−2】 图 5−20 为小车运输系统的示意图和 I/O 接线图，SQ1、SQ2、SQ3 和 SQ4 是限位开关，小车在 SQ1 处装料，10s 后右行，到 SQ2 后停止卸料 10s 后左行，碰到 SQ1 后停下装料，就这样不停循环工作，限位开关 SQ3 和 SQ4 的作用是当 SQ2 或者 SQ1 失效时，SQ3 和 SQ4 起保护作用，SB1 是起动按钮，SB2 是停止按钮。

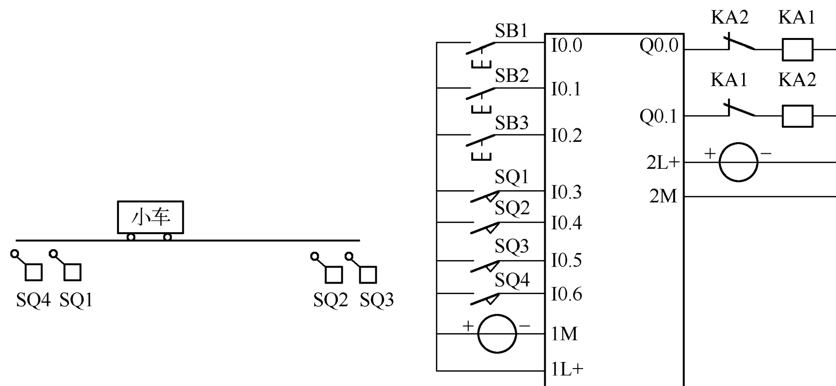


图 5−20 小车运输系统的示意图和 I/O 接线图

【解】

小车左行和右行是不能同时进行的，因此有联锁关系，与电动机的正、反转的梯形图类似，因此先画出电动机正、反转控制的梯形图，如图 5−21 所示，再在这个梯形图的基础上进行修改，增加 4 个限位开关的输入，增加两个定时器，就变成了图 5−22 的梯形图。

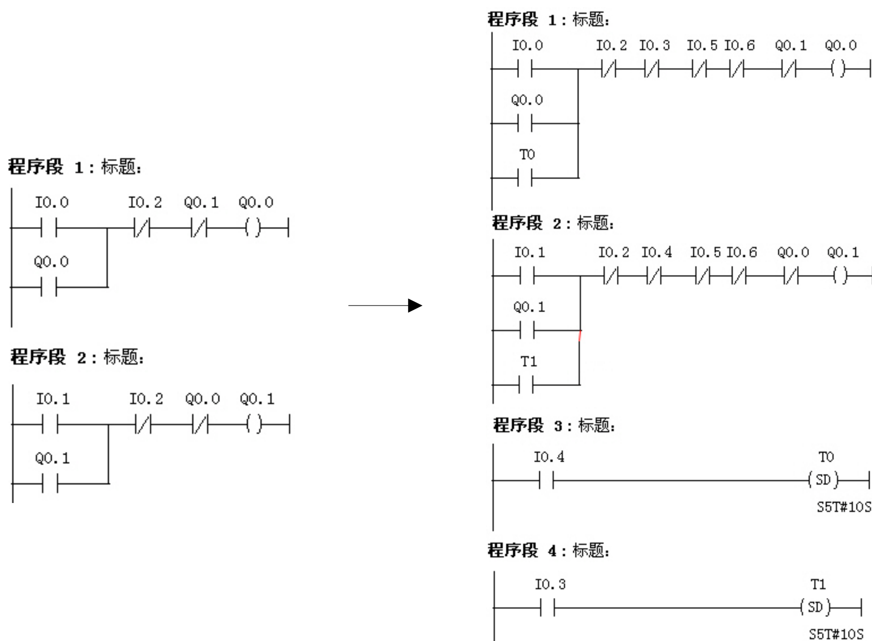


图 5−21 电动机正、反转控制的梯形图

图 5−22 小车运输系统的梯形图

5.2.2　功能图设计法

对于比较复杂的逻辑控制，用经验设计法就不合适，适合用功能图设计法。功能图设计法无疑是应用最为广泛的设计方法。功能图就是顺序功能图。功能图设计法就是先根据系统的控制要求画出功能图，再根据功能图设计梯形图。可采用基本指令、复位/置位指令和功能指令设计梯形图。因此，设计功能图是整个程序设计过程中的关键，也是难点。

1．利用基本指令编写梯形图指令

用基本指令编写梯形图指令是最容易被想到的方法，不需要了解较多的指令。采用这种方法编写程序的过程是：先根据控制要求设计正确的功能图，再根据功能图写出正确的布尔表达式，最后根据布尔表达式画基本指令梯形图。以下用一个例子讲解利用基本指令编写梯形图指令的方法。

【例 5–3】 如图 5–23 所示的气动机械手由 3 个汽缸组成，即汽缸 A、B、C。其接线图如图 5–24 所示。其工作过程是：当接近开关 SQ0 检测到有物体时，系统开始工作，汽缸 A 向左运行；到极限位置 SQ2 后，汽缸 B 向下运行，直到极限位置 SQ4 为止；接着手指汽缸 C 抓住物体，延时 1s；然后汽缸 B 向上运行；到极限位置 SQ3 后，汽缸 A 向右运行；到极限位置 SQ1，此时手指汽缸 C 释放物体，并延时 1s，完成搬运工作。电磁阀 YV1 上电汽缸 A 向左运行，电磁阀 YV2 上电汽缸 A 向右运行，电磁阀 YV3 上电，汽缸 B 向下运行，电磁阀 YV4 上电，汽缸 B 向上运行，电磁阀 YV5 上电，汽缸 C 夹紧，电磁阀 YV5 断电，汽缸 C 松开。请画出接线图、功能图和梯形图。

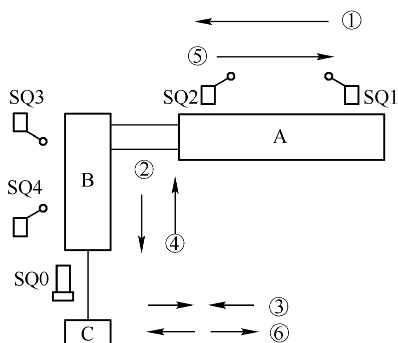


图 5–23　机械手示意图

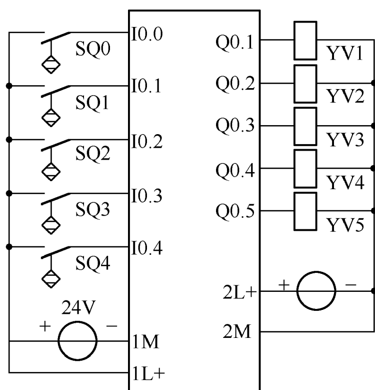


图 5–24　机械手接线图

【解】

这个运动逻辑看起来比较复杂，如果不掌握规律，则很难设计出正确的梯形图，一般先根据题意画出功能图，再根据功能图写出布尔表达式，如图 5–25 所示。布尔表达式是有规律的，当前步的步名对应的继电器（如 M0.1）等于上一步的步名对应的继电器（M0.0）与上一步的转换条件（I0.2）的乘积，再加上当前步的步名对应的继电器（M0.1）与下一步的步名对应的继电器非的乘积（$\overline{\text{M0.2}}$），其他的布尔表达式的写法类似，最后根据布尔表达式画出梯形图，如图 5–26 和图 5–27 所示。在整个过程中，功能图是关键，也是难点，而根据功能图写出布尔表达式和画出梯形图比较简单。



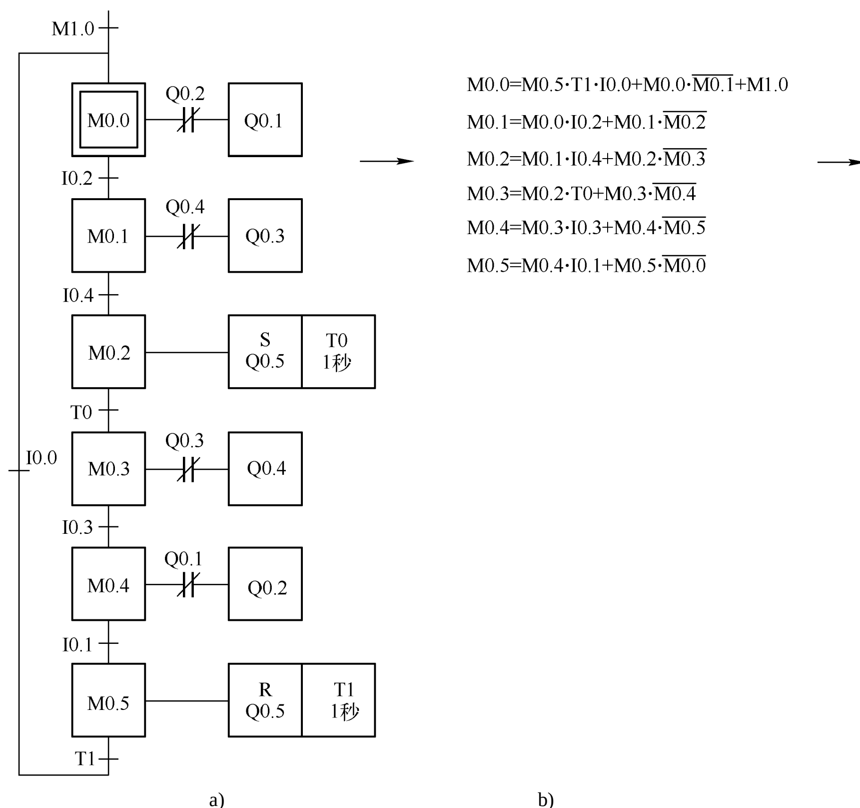


图 5–25　机械手的功能图和布尔表达式对应关系图

a) 功能图　b) 布尔表达式

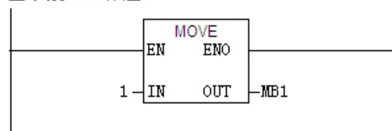


图 5–26　OB100 中的程序

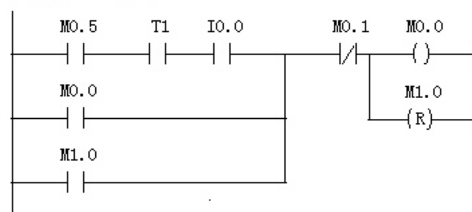
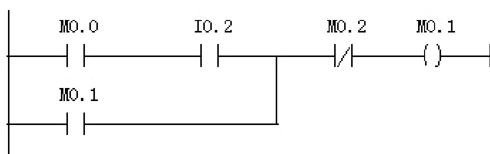
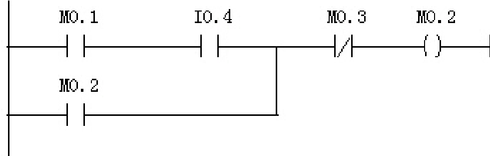


图 5–27　OB1 中的程序

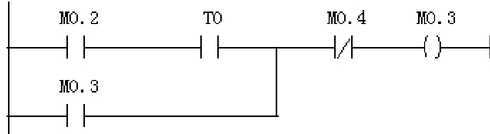
程序段 2：标题：



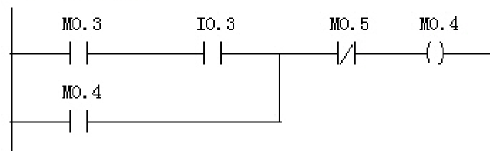
程序段 3：标题：



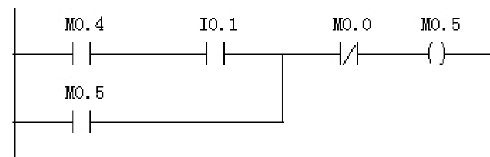
程序段 4：标题：



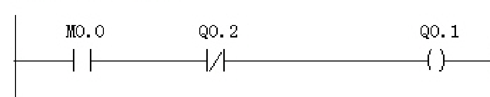
程序段 5：标题：



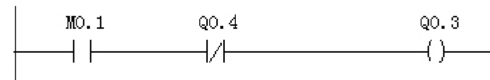
程序段 6：标题：



程序段 7：标题：



程序段 8：标题：



程序段 9：标题：

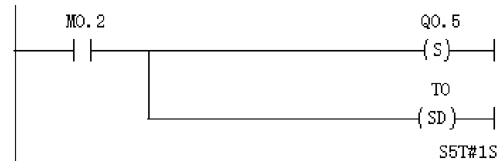
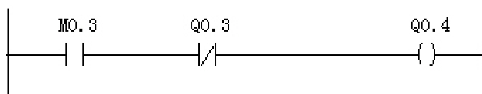


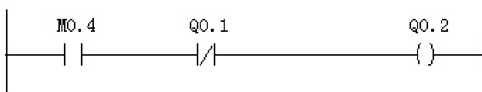
图 5-27　OB1 中的程序（续）



程序段 10：标题：



程序段 11：标题：



程序段 12：标题：

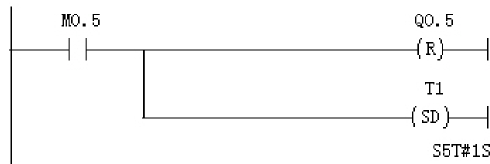


图 5-27　OB1 中的程序（续）

这个问题的解决方案仅从逻辑上讲是没有问题的，但解决方案中没有起动按钮，也没有复位和急停功能，因此是不符合实际的，也就是说没有实用价值。读者可以考虑如何改进以上方案。

2．利用功能指令编写逻辑控制程序

西门子的功能指令有许多特殊功能，其中移位指令和循环指令非常适合用于顺序控制，用这些指令编写程序简洁而且可读性强。以下用一个例子讲解利用功能指令编写逻辑控制程序。

【例 5-4】 用功能指令编写例 5-3 的程序。

【解】

梯形图如图 5-28 和图 5-29 所示。

程序段 1：标题：

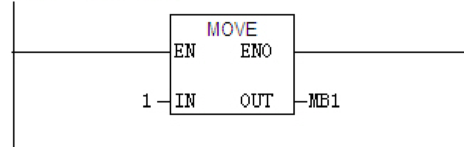
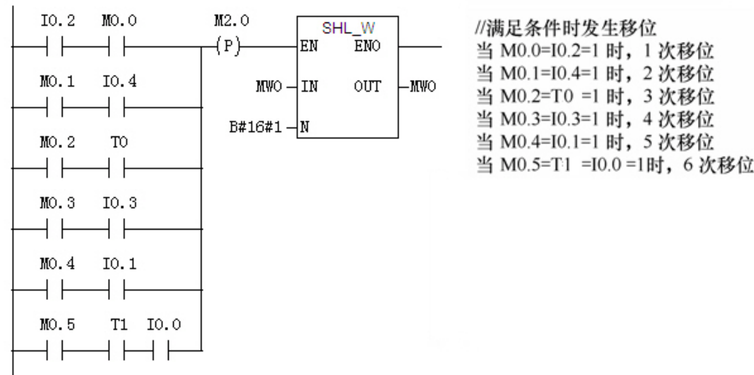
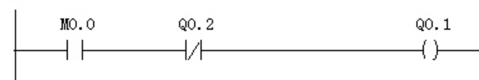


图 5-28　OB100 中的程序

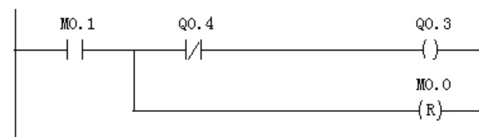
程序段 1：标题：



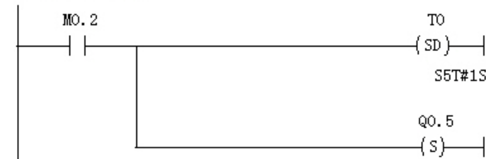
程序段 2：标题：



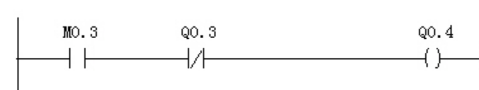
程序段 3：标题：



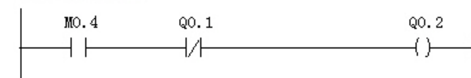
程序段 4：标题：



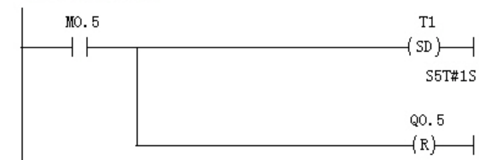
程序段 5：标题：



程序段 6：标题：



程序段 7：标题：



程序段 8：标题：

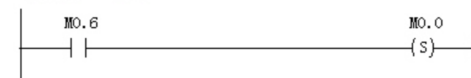


图 5-29　OB1 中的程序



3．利用复位和置位指令编写逻辑控制程序

复位和置位指令是常用指令，用复位和置位指令编写程序简洁而且可读性强。以下用一个例子讲解利用复位和置位指令编写逻辑控制程序。

【例 5–5】 用复位和置位指令编写例 5–3 的程序。

【解】

梯形图如图 5–31 和图 5–31 所示。

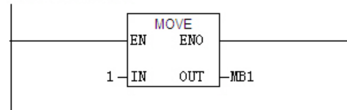
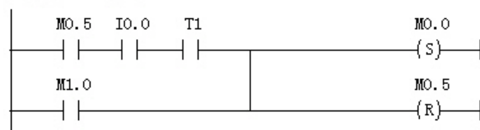
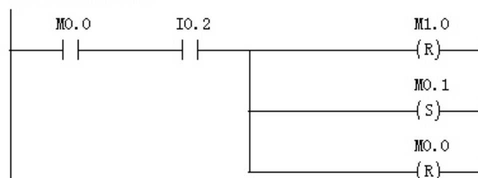
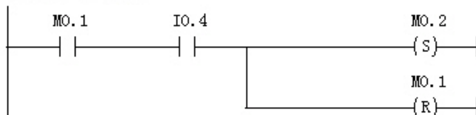
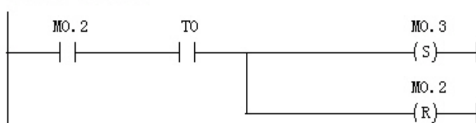


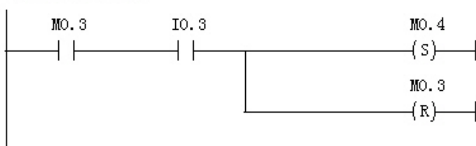
图 5–30 OB100 中的程序











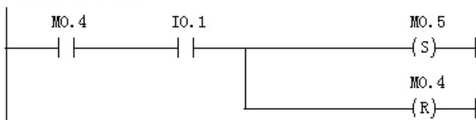
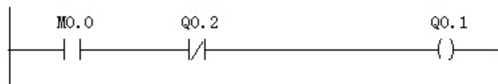
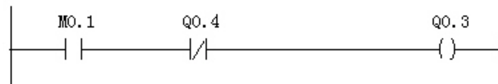


图 5–31 OB1 中的程序

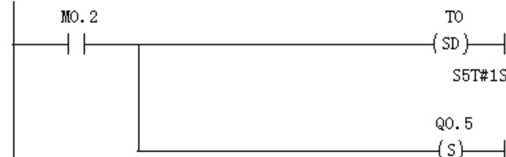
程序段 7：标题：



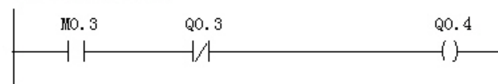
程序段 8：标题：



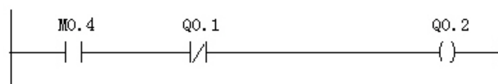
程序段 9：标题：



程序段 10：标题：



程序段 11：标题：



程序段 12：标题：

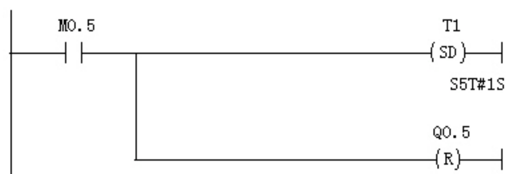


图 5-31　OB1 中的程序（续）

至此，同一个顺序控制的问题使用了基本指令、复位和置位指令和功能指令 3 种解决方案编写程序。3 种解决方案的编程都有各自几乎固定的步骤，但有一步是相同的，那就是首先都要画功能图。3 种解决方案没有好坏之分，读者可以根据自己的喜好选用。

5.3　S7-300/400 PLC的诊断与调试方法

STEP 7 提供了可视化的在线调试功能。在 STEP 7 中完成的硬件组态和用户程序必须下载到 PLC 中，经过软硬件的联合调试成功后，才能最终完成控制任务。

PLC 是运行在工业环境中的控制器，一般而言可靠性比较高，出现故障的概率较低，但出现故障也是难以避免的。一般引发故障的原因有很多，故障的后果也有很多种。

引发故障的原因虽然我们不能完全控制，但是我们可以通过日常的检查和定期的维护来消除多种隐患，把故障率降到最低。故障后果轻的可能造成设备的停机，影响生产的数量；重的可能造成财产损失和人员伤亡，如果是一些特殊的控制对象，一旦出现故障可能会引发更严重的后果。



对于维护人员来说最重要的是找到故障的原因，迅速排除故障，尽快恢复系统的运行。对于系统设计人员来说，在设计时要考虑到故障发生后系统的自我保护措施，力争使故障的停机时间最短，造成损失最小。

一般 PLC 的故障主要由外部故障或内部错误造成。外部故障由外部传感器或执行机构的故障等引发 PLC 产生故障，可能会使整个系统停机，甚至烧坏 PLC。

而内部错误是 PLC 内部的功能性错误或编程错误造成的，可以使系统停机。S7-300 PLC 具有很强的错误（或称故障）检测和处理能力，CPU 检测到某种错误后，操作系统调用对应的组织块，用户可以在组织块中编程，对发生的错误采取相应的措施。对于大多数错误，如果没有给组织块编程，出现错误时 CPU 将进入 STOP 模式。

5.3.1 使用状态和出错LED进行诊断

可以利用 CPU 面板上的指示灯进行初步的诊断，同时使用 STEP 7 软件的诊断功能进行诊断，快速查找故障原因。

状态和出错 LED 进行诊断是用于故障定位最直接、最简单的工具。要进一步确认故障位置，需要评估故障诊断缓冲区。S7-300 PLC 的指示灯与 CPU 状态关系见表 5−1。

表 5−1 S7-300 PLC 的指示灯与 CPU 状态关系

LED 状态（空白表示与此无关）					含义
SF	DC5V	FRCE	RUN	STOP	
关	关	关	关	关	CPU 无电源
关	开		关	开	CPU 处于 STOP 模式，正常状态
开	开		关	开	CPU 因出错处于 STOP 模式
	开		关	闪烁（0.5Hz）	CPU 请求存储器复位
	开		关	闪烁（2Hz）	CPU 正在执行存储器复位
	开		闪烁（2Hz）	开	CPU 正在处于启动状态
	开		闪烁（0.5Hz）	开	CPU 被编程设备的断电命令暂停
开	开				硬件或者软件错误
		开			启用“强制”功能
		闪烁（2Hz）			激活了节点闪烁测试
闪烁	闪烁	闪烁	闪烁	闪烁	CPU

5.3.2 使用STEP 7 的软件诊断功能进行硬件诊断

S7-300 PLC 具有非常强大的故障诊断功能，通过 STEP 7 编程软件可以获得大量的硬件故障与编程错误的信息，使用户能迅速地查找到故障。

这里的诊断是指 S7-300 PLC 内部集成的错误识别和记录功能，错误信息在 CPU 的诊断缓冲区内。有错误或事件发生时，标有日期和时间的信息被保存到诊断缓冲区，时间保存到系统的状态表中，如果用户已对有关的错误处理组织块编程，CPU 将调用该组织块。

1．故障诊断的基本方法

在 SIMATIC 管理器中用菜单命令“查看”→“在线”打开在线窗口。打开所有的站，查看是否有 CPU 显示了指示错误或故障的诊断符号。

诊断符号用来形象直观地表示模块的运行模式和模块的故障状态，如图 5-32 所示。如果模块有诊断信息，在模块符号上将会增加一个诊断符号或者模块符号的对比度降低。

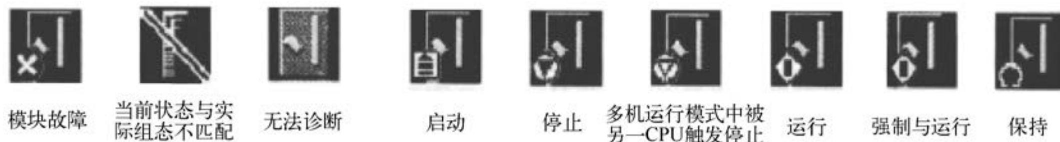


图 5-32　诊断符号

诊断符号“当前组态与实际组态不匹配”表示被组态的模块不存在或者插入了与组态的模块型号不同的模块。

诊断符号“无法诊断”表示无线上连接或该模块不支持模块诊断信息，例如电源模块或子模块。

“强制”符号表示在该模块上有变量被强制，即在模块的用户程序中有变量被赋予一个固定值，该数据值不能被程序改变。“强制”符号可以与其他符号组合在一起显示，如图 5-32 中“强制与运行”符号。

从在线的 SIMATIC 管理器的窗口、在线的硬件诊断功能打开的快速窗口和在线的硬件组态窗口（诊断窗口），都可以观察到诊断符号。

通过观察诊断符号，可以判断 CPU 模块的运行模式是否有强制变量，CPU 模块和功能模块（FM）是否有故障。

打开在线窗口，在 SIMATIC 管理器中执行菜单命令“PLC”→“诊断/设置”→“硬件诊断”，将打开硬件诊断快速浏览窗口。在该窗口中显示 PLC 的状态，看到诊断功能的模块硬件故障，双击“故障模块”可以获得详细的故障信息。

2．利用CPU诊断缓冲区进行详细故障诊断

建立与 PLC 的在线连接后，在 SIMATIC 管理器中选择要检查的站，执行菜单命令“PLC”→“诊断/设置”→“模块信息”，如图 5-33 所示，将打开“模块信息”窗口，显示该站中 CPU 的信息。在快速窗口中使用“模块信息”。

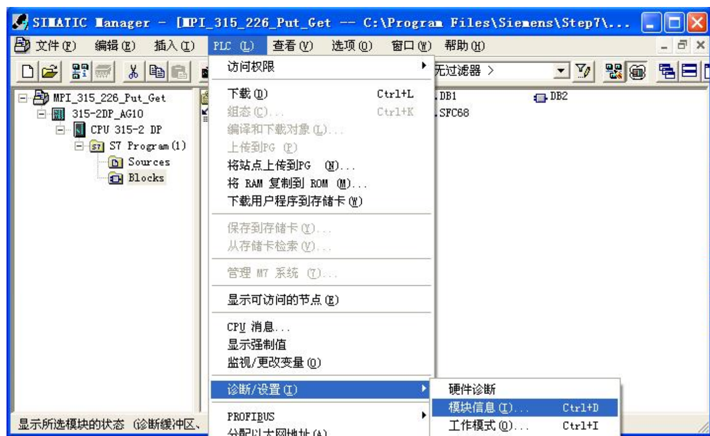


图 5-33　打开 CPU 诊断缓冲区



在“模块信息”窗口中的“诊断缓冲区（Diagnostic Buffer）”选项内，给出了CPU中发生的事件一览表，选中“事件”窗口中某一行的某一事件，下面灰色的“关于事件的详细资料”窗口将显示所选事件的详细信息，如图5-34所示。使用“诊断缓冲区”可以对系统的错误进行分析，查找停机的原因，并对出现的诊断时间分类。



图5-34　模块信息窗口

诊断事件包括模块故障、过程写错误、CPU中的系统错误、CPU运行模式的切换、用户程序的错误和用户用系统功能SFC52定义的诊断事件。

在“模块信息”窗口中，编号为“1”是位于最上面的事件，也是最近发生的事件。如果显示因编程错误造成CPU进入STOP模式，选择该事件，并单击“打开块（Open Block）”按钮，将在程序编辑器中打开与错误有关的块，显示出错的程序段。

诊断中断和DP从站诊断信息用于查找模块和DP从站中的故障原因。

“存储器（Memory）”选项给出了所选的CPU或M7功能模块的工作内存和装载内存当前的使用情况，可以检查CPU或功能模块的装载内存中是否有足够的空间用来存储新的块，如图5-35所示。

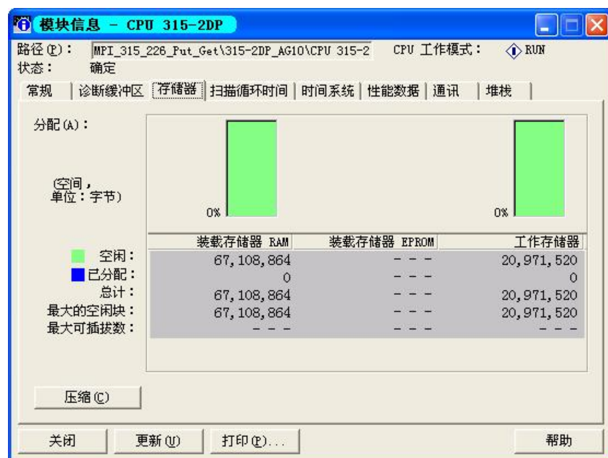


图5-35　“存储器（Memory）”选项




“扫描循环时间（Scan Cycle Time）”选项用于显示所选 CPU 或 M7 功能模块的最小循环时间、最大循环时间和当前循环时间，如图 5-36 所示。

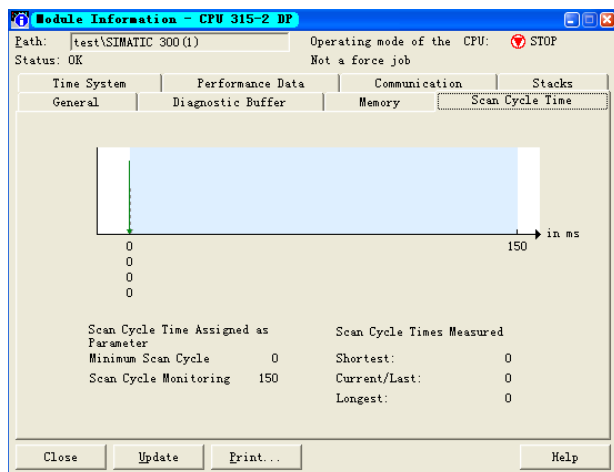


图 5-36　“扫描循环时间（Scan Cycle Time）”选项

如果最长循环时间接近组态的最大扫描循环时间，就会由于循环时间的波动可能产生时间错误，此时应增大设置的用户程序最大循环时间（监控时间）。

如果循环时间小于设置的最小循环时间，CPU 自动延长循环至设置的最小循环时间。在这个延长时间内可以处理背景组织块（OB90）。组态硬件时可以设置最大和最小循环时间。

3．错误处理组织块

组织块是操作系统与用户程序之间的接口。STEP 7 提供了各种不同的组织块（OB），用组织块可以创建在特定时间执行的程序和响应特定事件的程序。

系统程序可以检测下列错误：不正确的 CPU 功能、系统程序执行中的错误、用户程序中的错误和 I/O 中的错误。根据错误类型的不同，CPU 设置为进入 STOP 模式或调用一个错误处理 OB。具体可以参考第 4 章的相关内容。

5.3.3　用变量监控表进行调试

1．变量表的功能

变量表和 PLC 建立在线联系后，可以将硬件组态和程序下载到 PLC 中。用户可以通过 STEP 7 进行在线调试程序，寻找并发现程序设计中的问题。变量表上可以显示用户感兴趣的变量，它可以用于监视和修改变量值。变量表有如下功能：

1）监视变量。可以在编程设备上显示用户程序或 CPU 中每个变量值的当前值。

2）修改变量。可以将固定值赋给用户程序或 CPU 中的每个变量，使用程序状态测试时进行一次数值修改。

3）使用外部设备输出并激活修改值，允许在停机状态下将固定值赋给 CPU 的 I/O。

4）强制变量。可以为用户程序或 CPU 中的每个变量赋予一个固定值，这个值是不能被用户程序覆盖的。

用户可以显示或者赋值的变量包括：输入、输出、位存储、定时器、计数器、数据块的



内容和 I/O。

2．建立变量表

在管理器界面中，如图 5-37 所示。先选中“块”，再单击菜单栏中的“插入”→“S7 块”→“变量表”，弹出“属性—变量”界面，如图 5-38 所示，默认变量表为“VAT_1”，单击“确定”按钮便可生成变量表。

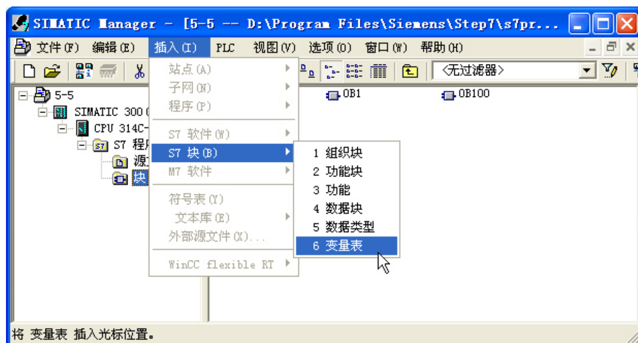


图 5-37　建立变量表

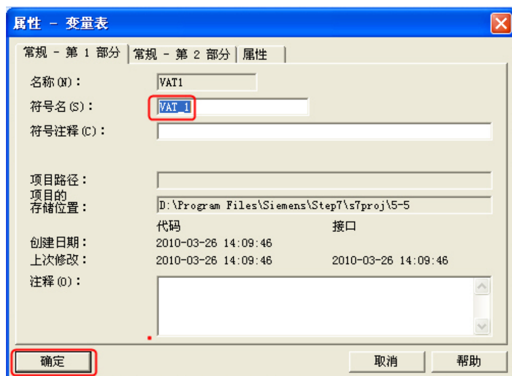


图 5-38　“属性—变量表”界面

回到管理器界面，双击“VAT_1”，弹出变量表界面，此时的变量表是空白的，并没有变量，如图 5-39 所示。

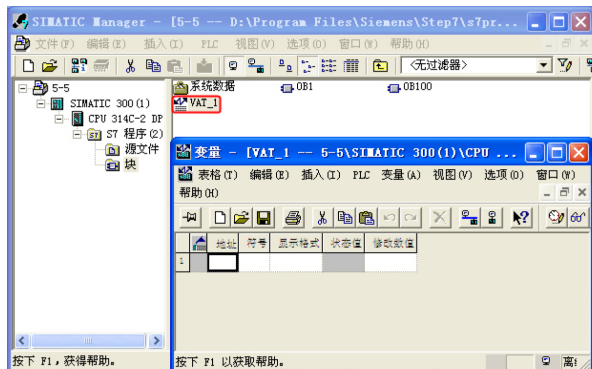


图 5-39　打开变量表





3．利用变量表调试程序

（1）输入变量

每个变量表中有 5 个栏，分别显示变量的 5 个属性：地址、符号、显示格式、状态值和修改值。一个变量表最多有 1024 行，每行最多可有 255 个字符。

用户可以通过在“符号”栏输入符号或在“地址”栏输入地址来插入变量，如果在符号表中已经定义地址相应的符号，则符号栏或者地址会自动输入，如图 5-40 所示。

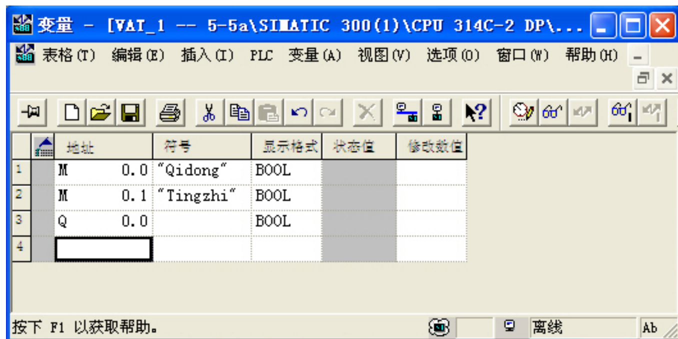


图 5-40　输入变量

（2）监视和修改变量

假设要调试的程序如图 5-41 所示，将整个工程下载到 CPU 中，注意一定要把变量表也下载到 CPU 中去。

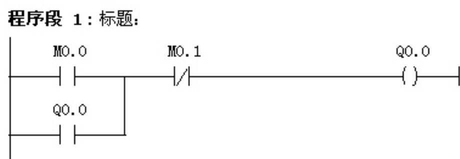


图 5-41　程序

1）变量表与 CPU 的连接。先建立变量表与 CPU 的连接，共有几种方法。单击菜单“PLC”→“连接到”，来定义与 CPU 的连接。子菜单有 3 个选项，如图 5-42 所示，第一个是组态的 CPU，其作用与单击工具栏中的按钮作用相同，用于建立被激活的变量表与 CPU 的连接；第二个是直接连接 CPU，其作用与单击工具栏中的按钮作用相同，用于直接连接 CPU（与编程设备用编程电缆连接的 CPU）之间的在线的连接；第三个是可访问的 CPU，在打开的对话框中，用户可以选择与哪个 CPU 建立连接。

使用菜单命令“PLC”→“断开连接”，可以断开变量表与 CPU 的连接。

2）变量表的监视。单击工具栏中的“监视变量”按钮，或者使用菜单中的“变量”→“监视”，便可监视程序中变量的情况，如图 5-43 所示，3 个变量的状态都有显示。

3）变量表的修改。当变量表处于监视状态时，在参数“M0.0”的“修改数值”栏中输入“true”（1 也可以），再单击“修改变量”按钮，可以看到参数“M0.0”为“true”，由于程序运行使得参数“Q0.0”也为“true”。当然也可以使用菜单中的“变量”→“修改”，来修改参数的数值。



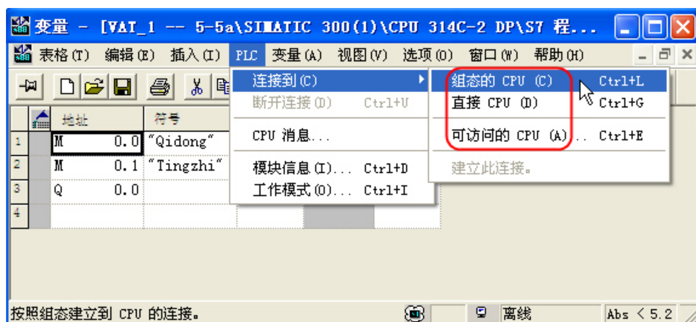


图 5-42　变量表与 CPU 的连接

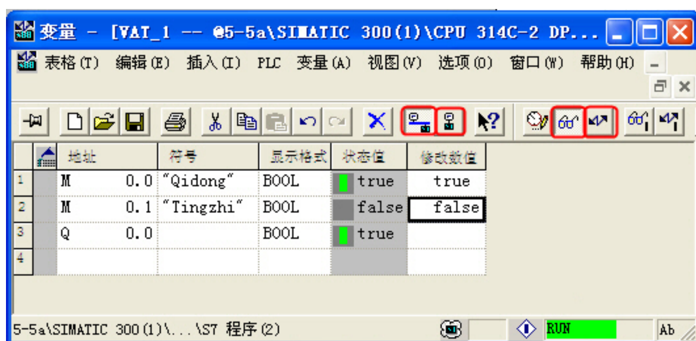


图 5-43　变量表的监视和修改

（3）强制变量

强制变量可以给用户一个固定值，它独立于程序运行，不会被执行的用户程序改变或者覆盖。强制的优点在于可以在不改变程序代码，也不改变硬件连线的情况下，强行改变输入和输出状态。

强制变量的方法是：先选中要强制变量中将要修改的数值，再使用菜单中的“变量”→“强制”即可。停止强制的方法是：使用菜单中的“变量”→“停止强制”。

5.3.4　使用PLCSIM软件进行调试

1．S7-PLCSIM简介

西门子为 S7-300/400 PLC 设计了一款可选仿真软件包 PLC Simulation（本书简称 S7-PLCSIM），此仿真软件包可以在计算机或者编程设备中模拟可编程序控制器运行和测试程序，它不能脱离 STEP 7 独立运行。如果 STEP 7 中已经安装仿真软件包，工具栏中的“仿真开关”按钮是亮色的，否则是灰色的，只有“仿真开关”按钮是亮色才可以用于仿真。

S7-PLCSIM 提供了简单的用户界面，用于监视和修改在程序中使用各种参数（如开关量输入和开关量输出）。当程序由 S7-PLCSIM 处理时，也可以在 STEP 7 软件中使用各种软件功能，如使用变量表监视、修改变量和断点测试功能。

2．S7-PLCSIM应用

S7-PLCSIM 仿真软件使用比较简单，以下用一个简单的例子介绍其使用方法。

【例 5-6】 将如图 5-44 所示的程序，用 S7-PLCSIM 进行仿真。



图 5-44　用于仿真的程序

【解】

具体步骤如下：

1）先新建一个工程，并进行硬件组态，在组织块 OB1 中输入如图 5-44 所示的程序，保存工程。

2）开启仿真。在管理界面中单击工具栏上的“仿真开关”按钮，如图 5-45 所示。

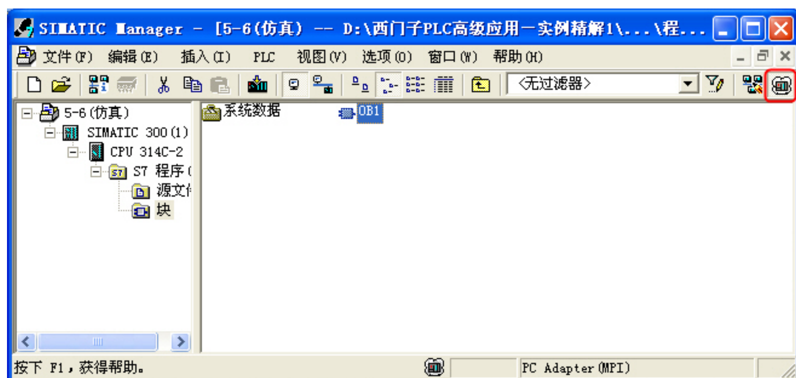


图 5-45　开启仿真

3）下载程序。先选定“SIMATIC 300（1）”，再单击工具栏的“下载”按钮，将硬件组态和程序下载到仿真器中，如图 5-46 所示。

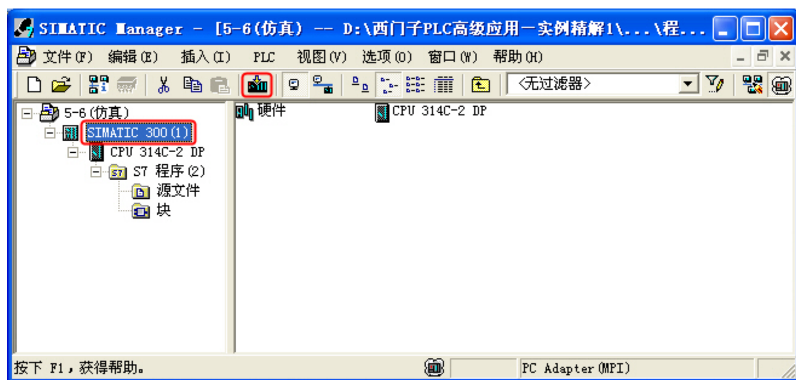


图 5-46　下载程序

4）进行仿真。先选择“RUN”，也就是将仿真器置于运行状态，再将 I0.0 上选取为“√”，也就是将 I0.0 置于“ON”，这时，Q0.0 也显示为“ON”；当去掉 I0.0 上“√”，也就是



将 I0.0 置于“OFF”，这时，Q0.0 上的“√”消失，即显示为“OFF”，如图 5-47 所示。

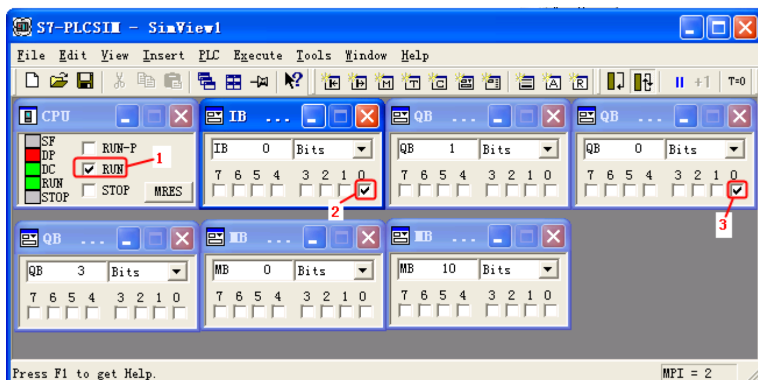


图 5-47 进行仿真

5）监视运行。打开程序编辑器，在工具栏中单击“监视”按钮 ，可以看到：若仿真器上的 I0.0 和 Q0.0 都是“ON”，则程序编辑器界面上的 I0.0 和 Q0.0 也都是“ON”，如图 5-48 所示。这个简单例子的仿真效果与下载程序到 PLC 中的效果基本相同，相比之下前者实施要容易得多。

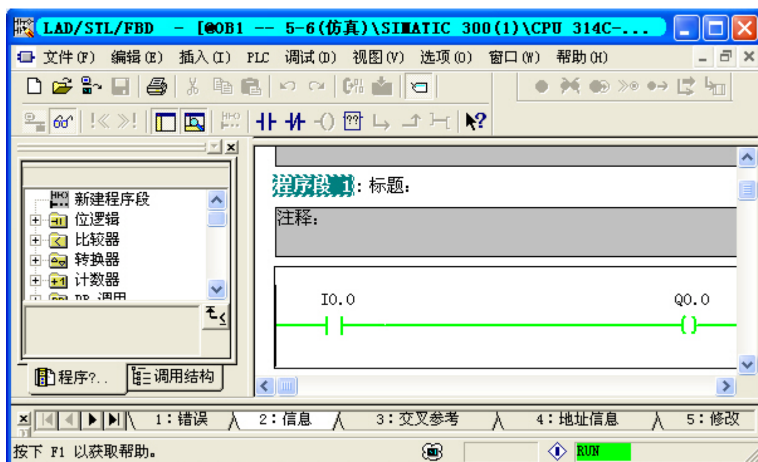


图 5-48 监视运行

3. S7-PLCSIM与真实PLC的差别

S7-PLCSIM 提供了方便、强大的仿真模拟功能。与真实的 PLC 相比，它的灵活性高，提供了许多 PLC 硬件无法实现的功能，使用也更加方便。但是仿真软件毕竟不能完全取代真实的硬件，不可能实现完全仿真。用户利用 S7-PLCSIM 进行仿真时，还应该了解它与真实 PLC 的差别。

（1）S7-PLCSIM 上有如下功能在真实 PLC 上无法实现

1）仿真的 CPU 中正在运行时可以用“Stop”选项中断程序，恢复“运行”时是从程序中断处开始继续处理程序。

2）与真实的 CPU 一样，仿真软件可以改变 CPU 的操作模式（RUN、RUN-P 和

STOP）。但与实际 CPU 不同的是仿真的 CPU 切换到 STOP 模式并不会改变输出的状态。

3）仿真软件中在目标视图中变量的每个改变，其存储区对应相关地址的内容会被同时更新。CPU 并不是等到循环周期结束或开始时才更新改变的数据。

4）使用关于程序处理的选项可以指定 CPU 如何执行程序：

- 选择“By cycles”程序执行一个周期后等待命令再执行下一个循环周期。
- 选择“Automatic”程序的处理同实际自控系统一样，一旦一个循环周期结束马上执行下一个周期。

5）仿真定时器可以使用自动或手动方式处理，自动方式按照程序执行结果，手动方式可以给定特殊值或复位定时器。复位定时器可以复位单独的定时器或一次复位所有定时器。

6）可以手动触发诊断中断 OB。OB40 到 OB47（过程中断）、OB70（I/O 冗余错误）、OB72（CPU 冗余错误）、OB73（通信冗余错误）、OB80（时间错误）、OB82（诊断警告）、OB83（插拔模块警告）、OB85（程序执行错误）和 OB86（机架故障）。

7）过程映像区和 I/O 区。如果改变一个输入映像区的值，S7-PLCSIM 立即将此值复制到输入外设区。这就意味着从输入外设区写到输入过程映像区所需要的值在下一个循环周期开始时不会丢失。同样如果改变了输出映像区的一个值，此值立即被复制到输出外设区。

（2）S7-PLCSIM 与“实际”的自动化系统还有以下不同

1）诊断缓冲区。S7-PLCSIM 不能支持所有写入诊断缓冲区的错误消息。例如，关于 CPU 中的电池电量不足的消息或者 EEPROM 错误是不能仿真的。但大部分 I/O 和程序错误都是可以仿真的。

2）在改变操作模式时（比如从 RUN 切换到 STOP）输入/输出没有“安全”状态。

3）不支持功能模块（FM）。

4）S7-PLCSIM 与 S7-400 PLC CPU 一样支持 4 个累加器。在某些情况下 S7-PLCSIM 上运行的程序与真实的只有两个累加器 S7-300 PLC CPU 上运行结果不同。

5）输入/输出的不同。大多数 S7-300 PLC 产品系列的 CPU 可以自动配置输入/输出设备。如果将模块连接到控制器，CPU 即自动地识别此模块。对于仿真的自动化系统，这种自动地识别是不能模拟的。如果把一个自动组态好 I/O 的 S7-300 PLC CPU 程序装载到 S7-PLCSIM 中，系统数据中将不包含任何 I/O 组态。因此，如果使用 S7-PLCSIM 来仿真 S7-300 PLC 的程序，为了使 CPU 能识别所使用的模块，必须首先装载硬件组态。在 S7-PLCSIM 中 S7-300 PLC CPU 不能自动识别 I/O，例如 CPU 315-2DP，CPU 316-2DP 或 CPU 318-2DP 等，为了能将硬件组态装载到 S7-PLCSIM，需要创建一个项目。复制相应的硬件组态到这个项目并装载到 S7-PLCSIM。然后从任意 STEP 7 项目装载程序块，I/O 处理都不会有错误。

此外，S7-PLCSIM V5.4 SP3 以前的版本不能对通信进行仿真。

5.4　实例

初学者在进行 PLC 控制系统的设计时，往往不知从何入手，其实 PLC 控制系统的设计有一个相对固定的模式，只要读者掌握了前述章节的知识，再按照这个模式进行，一般不难设计出正确的控制系统。以下用一个例子来说明 PLC 控制系统的设计过程。



【例 5-7】 有一台油压机，其液压系统如图 5-49 所示，其动作循环过程为：

1）当通电后 1YA 有电，液压缸实现快进。

2）当压下行程开关 SQ2 时，3YA 有电，回油流经调速阀 1 实现一工进。

3）当压下行程开关 SQ3 时，4YA 有电，回油流经调速阀 2 实现二工进。

4）当压下行程开关 SQ4 时，油缸开始压制物体，当无杆气腔油压力低于 4MPa 时，进行补压，当无杆气腔油压力高于 6MPa 时，5YA 有电卸荷，液压系统的油压高于 4MPa，并保持 5min 后，1YA 断电，2YA 有电，油路反向，实现油缸快退，当压下行程开关 SQ1 时，5YA 有电，系统卸荷。其中 SP1 压力继电器在 4MPa 时动作，SP2 压力继电器在 6MPa 时动作。

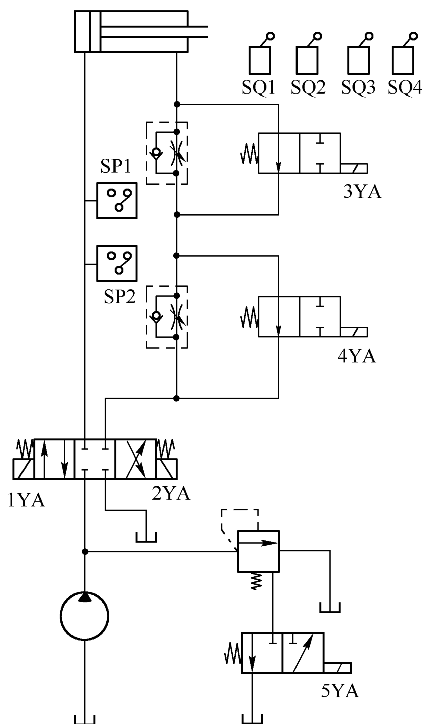


图 5-49 液压原理图

【解】

1．确定控制对象和控制范围

首先，要详细分析被控制对象、控制过程和要求。对于本例，控制对象就是液压系统，控制过程可以用表 5-2 表示。

表 5-2 电磁铁动作顺序

工步	1YA	2YA	3YA	4YA	5YA	输入信号
快进	+					SQ2
一工进	+		+			SQ3
二工进	+		+	+		SQ4

（续）

工步	1YA	2YA	3YA	4YA	5YA	输入信号
补压（与保压共 5min）	+		+	+		SQ4、SP1
保压（与补压共 5min）					+	SQ4、SP2
快退		+				SQ1
卸荷					+	复位

本例的输入/输出信号较多，而且逻辑过程也比较复杂，因此本液压系统适合用 PLC 控制。

2．PLC的选型

PLC 的种类繁多，不同种类的 PLC 功能差异很大，价格差距也很大，这为挑选 PLC 提供了很大的空间。机型选择的基本原则是：在功能满足要求的前提下，力争选用性能价格比高的，并且可以升级的产品。通常选型要注意以下几点：

（1）功能的选择

对于只有开关量和少量模拟量的控制设备，一般选用小型 PLC；对于有较多模拟量控制、有复杂的通信、多闭环控制系统，一般根据其规模选用中高档 PLC 机型；对于本例只有开关量，而且点数不超过 20 点，因此选用中小型机足够，选择面十分广泛，市面上的 PLC 应都能满足要求。

（2）基本单元的选择

基本单元的选择主要包括相应速度、结构形式和扩展能力。

PLC 输入信号与响应的输出信号之间存在着一定的时间延迟，称为延时时间，PLC 的延迟时间短则响应速度快。对于以开关量为主的系统，PLC 的响应速度能够满足要求，无需特殊考虑，而以模拟量为主的系统则必须考虑。本例只有开关量，因此不必考虑响应速度。

一般 PLC 有整体式和模块式两种结构形式，前者的价格便宜，而且通常小型 PLC 多采用整体式结构。

在扩展方面也应注意，如通信扩展，设计时不需要，并不代表以后用户一定不用。

（3）内存容量的估算

本系统有 SQ1、SQ2、SQ3、SQ4、SP1、SP2、起动 SB1、停止 SB1、复位 SB3 和急停 SB4 10 个输入信号，输出有 1YA、2YA、3YA、4YA、5YA、EL 和电动机的起动 KM 7 个信号。输入、输出信号共有 17 个，一般在选型时输入和输出点数要比实际多 20%～30%，富裕的点数通常预留给用户。所以输入、输出点数为 17×（1+25%）≈21，可以初步选定输入、输出点数和为 21。

由于西门子的 S7-300 PLC 性能可靠，故选定的型号为：CPU 313C（当然，读者也可以选用其他品牌的 PLC，如三菱的 FX 系列 PLC），此机型有 16 个输入，16 个输出。又因为本例的控制要求中没有提出通信等特殊要求，所以不需要选用特殊模块。

3．PLC控制系统的硬件设计

（1）PLC 的 I/O 分配

PLC 的 I/O 分配表见表 5-3。



表 5-3　PLC 的 I/O 分配表

输入			输出		
名称	符号	输入点	名称	符号	输出点
复位按钮	SB1	I0.0	电磁阀	1YA	Q0.0
起动按钮	SB2	I0.1	电磁阀	2YA	Q0.1
停止按钮	SB3	I0.2	电磁阀	3YA	Q0.2
限位开关	SQ1	I0.3	电磁阀	4YA	Q0.3
限位开关	SQ2	I0.4	电磁阀	5YA	Q0.4
限位开关	SQ3	I0.5	信号灯	EL	Q0.5
限位开关	SQ4	I0.6	接触器	KM	Q0.7
压力继电器	SP1	I0.7			
压力继电器	SP2	I1.0			
急停按钮	SB4	I1.1			

（2）PLC 的硬件系统设计时要注意以下问题

① 西门子 S7-300 PLC 输入端需外接 DC 24V 电源，可以选用西门子生产的电源，但西门子的电源价格较高，使用市面上的品牌开关电源是完全能够满足要求的。这点有别于三菱 FX 系列 PLC，三菱 FX 系列 PLC 的输入端不要外接电源。

② 输入端开关（如按钮）一般都用常开触点，但注意急停开关必须接常闭触点，尽管从逻辑上讲，急停开关接常开触点是可行的，但如果急停开关接常开触点，由于某些情况下，急停开关的连线断开，系统仍然能正常工作，但紧急情况下，特别是重大事故发生时，急停开关将不能起到作用，很容易发生事故。如果急停开关接常闭触点，只要急停开关的连线断开，系统就不能正常工作。因此，“急停开关”总是能起到作用。

③ 输出端若用直流电源，则最好与接触器或继电器线圈并联一个续流二极管，续流二极管起续流作用。读者应注意二极管的接法，若接反，则会短路。

④ 输出端若用交流电源，则最好与接触器或继电器线圈并联一个浪涌吸收器。

⑤ 接触器的电源是 220V 交流电，而电磁阀的电源是 24V 直流电，因此二者一定不能共用一组输出。

⑥ 在画 I/O 接线图前务必认真阅读选定型号 PLC 的使用说明书，否则极易出错。

⑦ 由于 PLC 的继电器比较小，允许通过的电流只有 2A，损坏后不易更换，因此建议将每个输出点转接一个中间继电器，再由中间继电器控制电磁阀和接触器的线圈，尽量保护 PLC。

4．PLC控制系统的软件设计

先画 I/O 接线图，如图 5-50 所示，再画功能图，如图 5-51 所示，根据功能图画程序图，如图 5-52～图 5-53 所示。

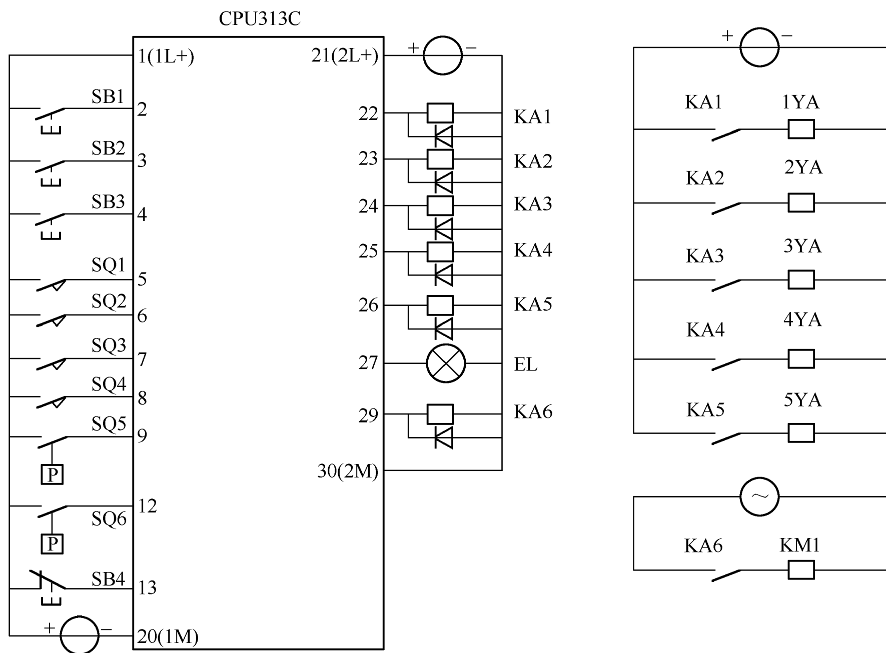


图 5-50　I/O 接线图

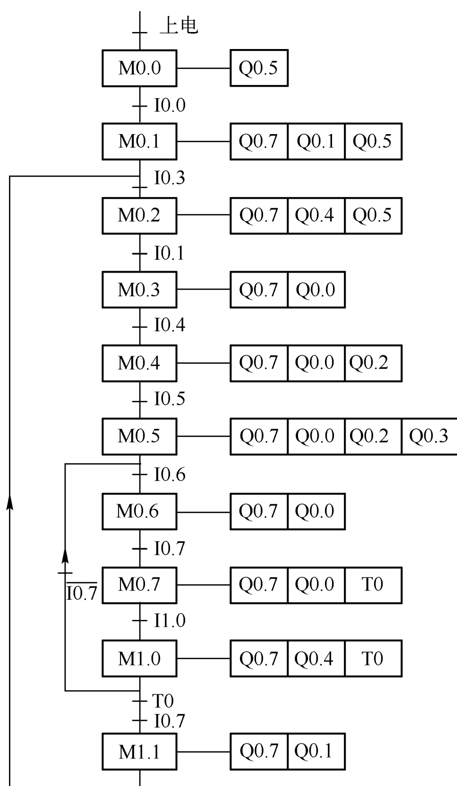


图 5-51　功能图



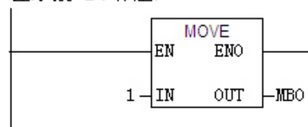
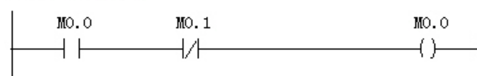
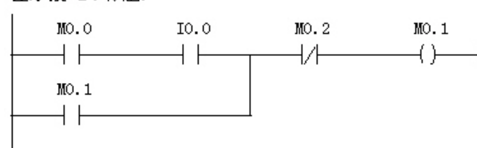
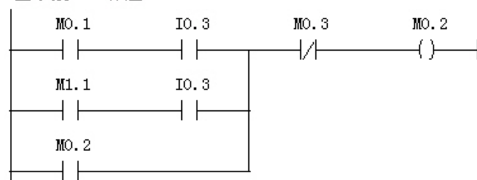
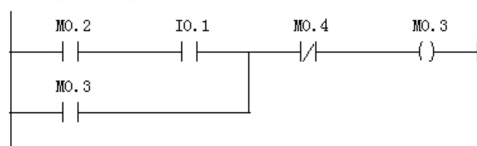


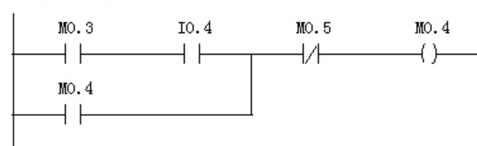
图 5–52　OB100 中的程序

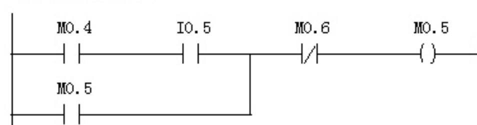












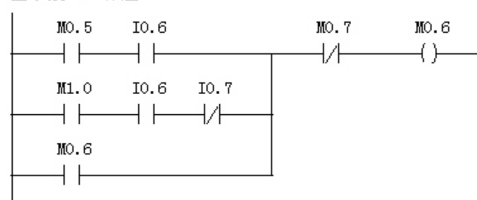
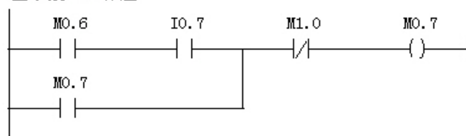
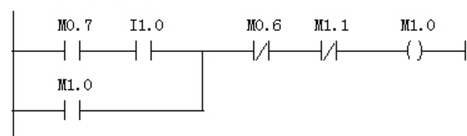


图 5–53　OB1 中的程序

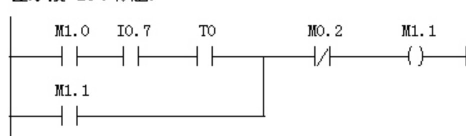
程序段 8：标题：



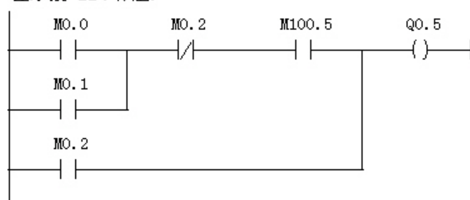
程序段 9：标题：



程序段 10：标题：



程序段 11：标题：



程序段 12：标题：



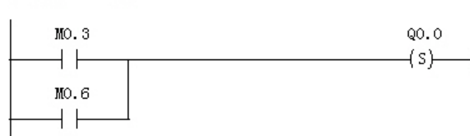
程序段 13：标题：



程序段 14：标题：



程序段 15：标题：



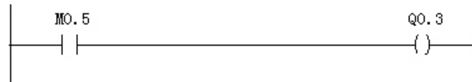
程序段 16：标题：



图 5-53　OB1 中的程序（续）



程序段 17：标题：



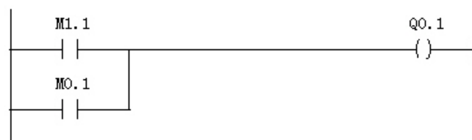
程序段 18：标题：



程序段 19：标题：



程序段 20：标题：



程序段 21：标题：



程序段 22：标题：

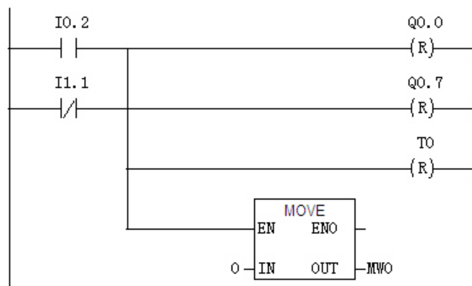


图 5-53　OB1 中的程序（续）

小结

重点难点总结

本章的内容比较多，重难点内容多，读者必须掌握的内容有：

1．要学会设计功能图，这是非常重要的。因为比较复杂的程序都要根据功能图编写，因此如果不能根据工程的要求设计出功能图，则很难编写正确的程序。

2．程序的编写方法，同一个顺序控制的问题使用了基本指令、复位和置位指令和功能指令 3 种解决方案编写程序。3 种解决方案的编程都有各自的几乎固定的步骤，但有一步是相同的，那就是首先都要画功能图。3 种解决方案没有好坏之分，读者可以根据自己的喜好选用。相对而言，用置位/复位指令编写程序更加容易被初学者掌握。

3．S7-300/400 PLC 的功能比较强大。要学会使用 STEP 7 的软件诊断功能进行硬件诊断。

4．STEP 7 的调试功能也比较强大，应该重点掌握。

5．在没有硬件的情况下，验证 S7-300/400 PLC 程序正确性最好的办法是用仿真软件 S7-PLCSIM，随着软件的升级，其功能越来越强大，S7-PLCSIM V5.4SP3 已经具备通信仿真功能。

习题

1．在功能图中，什么是步、活动步、动作和转换条件？

2．设计功能图要注意什么？

3．编写梯形图程序要注意哪些问题？

4．编写逻辑控制梯形图程序有哪些常用的方法？

5．用 I0.0 控制 Q0.0、Q0.1 和 Q0.2，要求：I0.0 闭合两次，Q0.0 亮，I0.0 再闭合两次，Q0.1 亮，I0.0 再闭合两次，Q0.2 亮，I0.0 再闭合一次，Q0.0、Q0.1 和 Q0.2 灭，如此循环，请先设计功能图和接线图，再编写程序，要求用基本指令、移位指令和复位/置位指令编写。

6．要求用移位指令使 8 盏灯以 0.2s 的速度自左向右亮起，到达最右侧后，再自右向左回到最左侧。如此循环，I0.0 按钮为起动按钮，I0.1 为停止按钮，请先设计功能图和接线图，再编写程序，要求用基本指令、移位指令和复位/置位指令编写。

7．正反转的程序如图 5-54 所示，请用仿真软件 S7-PLCSIM 进行仿真，验证程序是否正确？

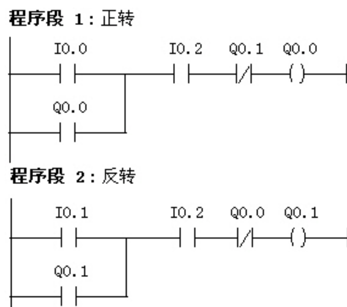


图 5-54　程序

8．S7-PLCSIM 与“实际”的自动化系统有什么不同？

9．根据如图 5-55 所示的功能图编写程序。

10．根据如图 5-56 所示的功能图编写程序。

11．机械手的工作示意图如图 5-57 所示，当合上按钮 I0.4，机械手将工件从 A 点搬运到 B 点，然后返回 A 点，再将如此循环，任何时候合上按钮 I0.5 时系统复位回到原点，请编写控制程序。



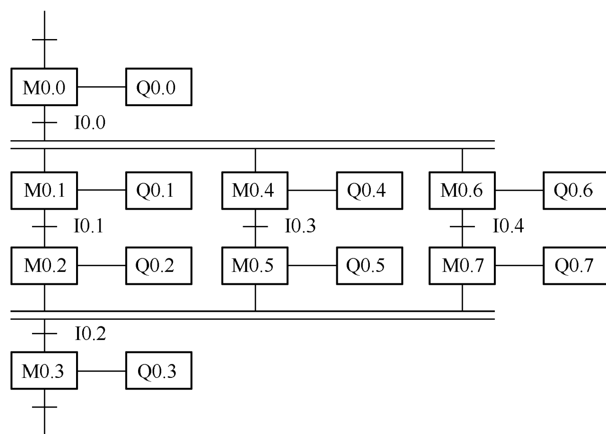


图 5-55　功能图

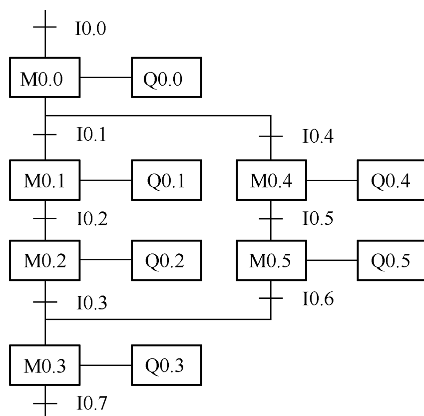


图 5-56　功能图

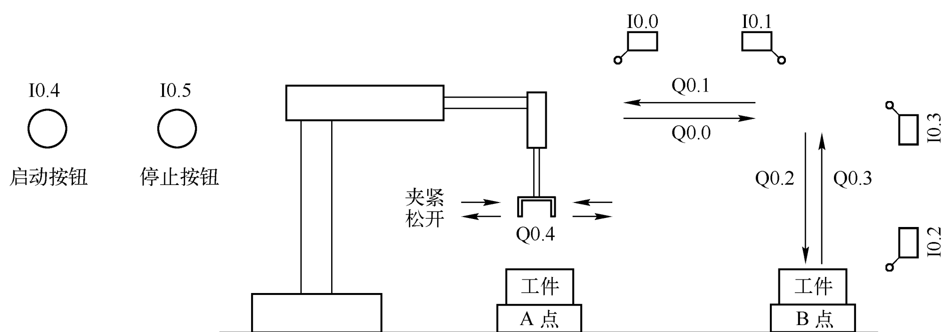


图 5-57　机械手示意图

第6章

S7-300/400 PLC的通信及其应用

本章介绍 S7-300 PLC 可编程序控制器的通信基础知识，并用实例介绍 S7-300 PLC 与 S7-200 PLC、S7-300 PLC 与 S7-300/400 PLC 之间的 MPI 通信；S7-300 PLC 与 S7-200 PLC、S7-300 PLC 与 S7-300/400 PLC 之间的 PROFIBUS 通信；S7-300 PLC 与 S7-200 PLC、S7-300 PLC 与 S7-300/400 PLC 之间的以太网通信。

6.1　通信基础知识

PLC 的通信包括 PLC 与 PLC 之间的通信、PLC 与上位计算机之间的通信以及和其他智能设备之间的通信。PLC 与 PLC 之间通信的实质就是计算机的通信，使得众多独立的控制任务构成一个控制工程整体，形成模块控制体系。PLC 与计算机连接组成网络，将 PLC 用于控制工业现场，计算机用于编程、显示和管理等任务，构成“集中管理、分散控制”的分布式控制系统（DCS）。

6.1.1　通信的基本概念

1．串行通信与并行通信

串行通信和并行通信是两种不同的数据传输方式。

串行通信就是通过一对导线将发送方与接收方进行连接，传输数据的每个二进制位，按照规定顺序在同一导线上依次发送与接收。例如，常用的优盘 USB 接口就是串行通信。串行通信的特点是通信控制复杂，通信电缆少，因此与并行通信相比，成本低。

并行通信就是将一个 8 位数据（或 16 位、32 位）的每一个二进制位采用单独的导线进行传输，并将传送方和接收方进行并行连接，一个数据的各二进制位可以在同一时间内一次传送。例如，老式打印机的打印口和计算机的通信就是并行通信。并行通信的特点是一个周期里可以一次传输多位数据，其连线的电缆多，因此长距离传送时成本高。

2．异步通信与同步通信

异步通信与同步通信也称为异步传送与同步传送，这是串行通信的两种基本信息传送方式。从用户的角度上说，两者最主要的区别在于通信方式的“帧”不同。

异步通信方式又称起止方式。它在发送字符时，要先发送起始位，然后是字符本身，最后是停止位，字符之后还可以加入奇偶校验位。异步通信方式具有硬件简单、成本低的特点，主要用于传输速率低于 19.2Kbit/s 以下的数据通信。

同步通信方式在传递数据的同时，也传输时钟同步信号，并始终按照给定的时刻采集数据。其传输数据的效率高，硬件复杂，成本高，一般用于传输速率高于 20Kbit/s 以上的数据通信。



3．单工、全双工与半双工

单工、双工与半双工是通信中描述数据传送方向的专用术语。

1）单工（Simplex）：指数据只能实现单向传送的通信方式，一般用于数据的输出，不可以进行数据交换。

2）全双工（Full Simplex）：也称双工，指数据可以进行双向数据传送，同一时刻既能发送数据，也能接收数据。通常需要两对双绞线连接，通信线路成本高。例如，RS-422 就是“全双工”通信方式。

3）半双工（Half Simplex）：指数据可以进行双向数据传送，同一时刻，只能发送数据或者接收数据。通常需要一对双绞线连接，与全双工相比，通信线路成本低。例如，RS-485只用一对双绞线时就是“半双工”通信方式。

4．PLC网络的术语解释

PLC 网络中的名词、术语很多，现将常用的予以介绍。

1）站（Station）：在 PLC 网络系统中，将可以进行数据通信、连接外部输入/输出的物理设备称为“站”。例如，由 PLC 组成的网络系统中，每台 PLC 可以是一个“站”。

2）主站（Master Station）：PLC 网络系统中进行数据连接的系统控制站，主站上设置了控制整个网络的参数，每个网络系统只有一个主站，主站号固定为“0”，站号实际就是 PLC 在网络中的地址。

3）从站（Slave Station）：PLC 网络系统中，除主站外，其他的站称为“从站”。

4）远程设备站（Remote Device Station）：PLC 网络系统中，能同时处理二进制位、字的从站。

5）本地站（Local Station）：PLC 网络系统中，带有 CPU 模块并可以与主站以及其他本地站进行循环传输的站。

6）站数（Number of Station）：PLC 网络系统中，所有物理设备（站）所占用的“内存站数”的综合。

7）网关（Gateway）：又称网间连接器、协议转换器。网关在传输层上以实现网络互联，是最复杂的网络互联设备，仅用于两个高层协议不同的网络互联。网关的结构也和路由器类似，不同的是互联层。网关既可以用于广域网互联，也可以用于局域网互联。网关是一种充当转换重任的计算机系统或设备。在使用不同的通信协议、数据格式或语言，甚至体系结构完全不同的两种系统之间，网关是一个翻译器。例如 AS-I 网络的信息要传送到由西门子 S7-200 PLC 组成的 PPI 网络，就要通过 CP243-2 通信模块进行转换，这个模块实际上就是网关。

8）中继器（Repeater）：用于网络信号放大、调整的网络互联设备，能有效延长网络的连接长度。例如，以太网的正常传送距离是 500m，经过中继器放大后，可传输 2500m。由于存在损耗，在线路上传输的信号功率会逐渐衰减，衰减到一定程度时将造成信号失真，因此会导致接收错误。中继器就是为解决这一问题而设计的。它完成物理线路的连接，对衰减的信号进行放大，保持与原数据相同。一般情况下，中继器的两端连接的是相同的媒体，但有的中继器也可以完成不同媒体的转接工作。

9）网桥（Bridge）：网桥将两个相似的网络连接起来，并对网络数据的流通进行管理。网桥的功能在延长网络跨度上类似于中继器，然而它能提供智能化连接服务，即根据帧的终

点地址处于哪一网段来进行转发和滤除。

10）路由器（Router，转发者）：所谓路由就是指通过相互连接的网络 把信息从源地点移动到目标地点的活动。一般来说，在路由过程中，信息至少会经过一个或多个中间节点。路由器是互联网的主要节点设备。路由器通过路由决定数据的转发。转发策略称为路由选择（routing），这也是路由器名称的由来。作为不同网络之间互相连接的枢纽，路由器系统构成了基于TCP/IP的国际互联网络Internet的主体脉络，也可以说，路由器构成了Internet的骨架。它的处理速度是网络通信 的主要瓶颈之一，它的可靠性则直接影响着网络互连的质量。因此，在园区网、地区网、乃至整个Internet研究领域中，路由器技术始终处于核心地位，其发展历程和方向，成为整个Internet研究的一个缩影。

11）交换机（Switch）：交换机是一种基于 MAC 地址识别，能完成封装转发数据包功能的网络设备。交换机可以"学习"MAC 地址，并把其存放在内部地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地址。

交换机通过直通式、存储转发和碎片隔离 3 种方式进行交换。

交换机的传输模式有全双工、半双工、全双工/半双工自适应。

6.1.2　RS-485 标准串行接口

1．RS-485 接口

RS-485 接口是在 RS-422 基础上发展起来的一种 EIA 标准串行接口，采用"平衡差分驱动"方式。RS-485 接口满足 RS-422 的全部技术规范，可以用于 RS-422 通信。RS-485 接口通常采用 9 针连接器。RS-485 接口的引脚功能参见表 6-1。

表 6-1　RS-485 接口的引脚功能

PLC 侧引脚	信 号 代 号	信 号 功 能
1	SG 或 GND	机壳接地
2	+24V 返回	逻辑地
3	RXD+或 TXD+	RS-485 的 B，数据发送/接收+端
5	+5V 返回	逻辑地
6	+5V	+5V
7	+24V	+24V
8	RXD－或 TXD－	RS-485 的 A，数据发送/接收－端
9	不适用	10 位协议选择（输入）

2．西门子的PLC连线

西门子 PLC 的 PPI 通信、MPI 通信和 PROFIBUS-DP 现场总线通信的物理层都是 RS-485，而且采用都是相同的通信线缆和专用网络接头。西门子提供两种网络接头，即标准网络接头和包括编程端口接头，可方便地将多台设备与网络连接，编程端口允许用户将编程站或 HMI 设备与网络连接，而不会干扰任何现有网络连接，图 6-1 为带编程口的网络接头。标准网络接头的编程端口接头均有两套终端螺丝钉，用于连接输入和输出网络电缆。这两种接头还配有开关，可选择网络偏流和终端。图 6-1 显示了电缆接头的普通偏流和终端状况，右端的电阻设置为"on"，而中间的设置为"off"，图中只显示了一个，若有多个也是这样设



置。要将偏流电阻设置“on”或者“off”，只要拨动网络接头上的拨钮即可。图 6-1 中拨钮在“off”一侧，因此偏置电阻未接入电路。

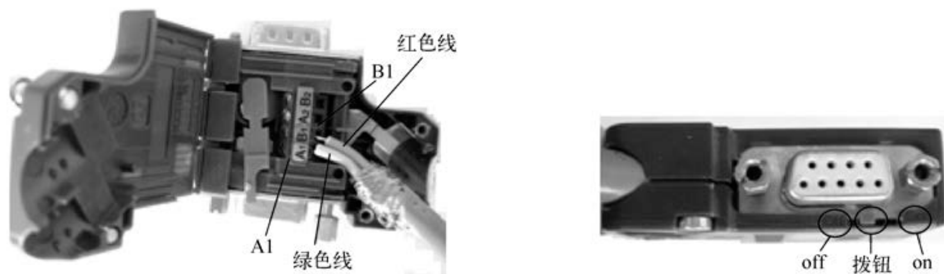


图 6-1　网络接头的偏流电阻设置图

【关键点】西门子的专用 PROFIBUS 电缆中有两根线，一根为红色，上标有“B”，一根为绿色，上面标有“A”，这两根线只要与网络接头上相对应的“A”和“B”接线端子相连即可（如“A”线与“A”接线端相连）。网络接头直接插在 PLC 的通信口上即可，不需要其他设备。注意：三菱的 FX 系列 PLC 的 RS-485 通信要加 RS-485 专用通信模块和偏流电阻。

6.1.3　OSI参考模型

通信网络的核心是 OSI（OSI-Open System Interconnection，开放式系统互联）参考模型。为了理解网络的操作方法，为创建和实现网络标准、设备和网络互联规划提供了一个框架。1984 年，国际标准化组织（ISO），提出了开放式系统互联的 7 层模型，即 OSI 模型。该模型自下而上分为：物理层、数据链接层、网络层、传输层、会话层、表示层和应用层。理解 OSI 参考模型比较难，但了解它，对掌握后续的以太网通信和 PROFIBUS 通信是很有帮助的。

OSI 的上 3 层通常称为应用层，用来处理用户接口、数据格式和应用程序的访问。下 4 层负责定义数据的物理传输介质和网络设备。OSI 参考模型定义了大多数协议栈共有的基本框架，如图 6-2 所示。

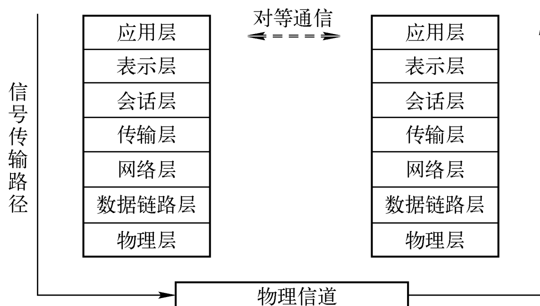


图 6-2　信息在 OSI 模型中的流动形式

1）物理层（Physical Layer）：定义了传输介质、连接器和信号发生器的类型，规定了物理连接的电气、机械功能特性，如电压、传输速率、传输距离等特性。典型的物理层设备有集线器（HUB）和中继器等。

2）数据链路层（Data Link Layer）：确定传输站点物理地址以及将消息传送到协议栈，

提供顺序控制和数据流向控制。该层可以继续分为两个子层：介质访问控制层（Mediuum Access Control，MAC）和逻辑链路层（Logical Link Control Layer，LLC），即层 2a 和 2b。其中 IEEE802.3（Ethernet，CSMA/CD）就是 MAC 层常用的通信标准。典型的数据链路层的设备有交换机和网桥等。

3）网络层（Network Layer）：定义了设备间通过逻辑地址（IP-Internet Protocol 因特网协议地址）传输数据，连接位于不同广播域的设备，常用来组织路由。典型的网络层设备是路由器。

4）传输层（Transport Layer）：建立会话连接，分配服务访问点（Sevice Access Point，SAP），允许数据进行可靠（Transmission Control Protocol，TCP 传输控制协议）或者不可靠（User Datagram Protocol，UDP 用户数据报协议）的传输。可以提供通信质量检测服务（QOS）。网关是互联网设备中最复杂的，它是传输层及以上层的设备。

5）会话层（Session Layer）：负责建立、管理和终止表示层实体间通信会话，处理不同设备应用程序间的服务请求和响应。

6）表示层（Presentation Layer）：提供多种编码用于应用层的数据转化服务。

7）应用层（Application Layer）：定义用户及用户应用程序接口与协议对网络访问的切入点。目前各种应用版本较多，很难建立统一的标准。在工控领域常用的标准是（Mutimedia Messaging Service，MMS 多媒体信息服务），用来描述制造业应用的服务和协议。

数据经过封装后通过物理介质传输到网络上，接收设备除去附加信息后，将数据上传到上层堆栈层。

各层的数据单位一般有各自特定的称呼。物理层的单位是比特（bit）；数据链路层的单位是帧（frame）；网络层的单位是分组（packet，有时也称包）；传输层的单位是数据报（datagram）或者段（segment）；会话层、表示层和应用层的单位是消息（message）。

6.2　MPI通信及其应用

6.2.1　MPI通信简介

MPI 网络可用于单元层，它是多点接口（Multi Point Interface）的简称，是西门子公司开发的用于 PLC 之间通信的保密的协议。MPI 通信是当通信速率要求不高、通信数据量不大时，可以采用的一种简单经济的通信方式。

主要的优点是 CPU 可以同时与多个设备建立通信联系。也就是说，编程器、HMI 设备和其他的 PLC 可以连接在一起并同时运行。编程器通过 MPI 接口生成的网络还可以访问所连接硬件站上的所有智能模块。可同时连接的其他通信对象的数目取决于 CPU 的型号。例如，CPU 314 的最大连接数为 4，CPU 416 为 64。

MPI 接口的主要特性为：

- RS-485 物理接口。
- 传输率为 19.2Kbit/s 或 187.5 Kbit/s 或 1.5Mbit/s。
- 最大连接距离为 50m（2 个相邻节点之间），有两个中继器时为 1100m，采用光纤和星形偶合器时为 23.8km。



- 采用 PROFIBUS 元件（电缆、连接器）。

MPI 通信有全局数据通信、基本通信和扩展通信，以下将分别介绍。

① 全局数据通信，这种通信方法通过 MPI 接口在 CPU 间循环地交换数据，而不需要编程。当过程映像被刷新时，在循环扫描检测点上进行数据交换。对于 S7-400 PLC，数据交换可以用 SFC 来启动。全局数据可以是输入、输出、标志位、定时器、计数器和数据块区。

数据通信不需要编程，而是利用全局数据表来配置。不需要 CPU 的连接用于全局数据通信。

② 基本通信，这种通信方法可用于所有 S7-300/400 PLC CPU，它通过 MPI 子网或站中的 K 总线来传送数据。系统功能（SFC），例如 X_SEND（在发送端）和 X_RCV（在接收端）被用户程序调用。最大用户数据量为 76 B。当系统功能被调用时，通信连接被动态地建立和断开。在 CPU 上需要有一个自由的连接。

③ 扩展通信，这种通信方法可用于所有的 S7-400 PLC CPU。通过任何子网（MPI，Profibus, Industrial Ethernet）可以传送最多 64KB 的数据。它是通过系统功能块（SFB）来实现的，支持有应答的通信。数据也可以读出或写入到 S7-300 PLC（PUT/GET 块）。不仅可以传送数据，而且可以执行控制功能，例如控制通信对象的起动和停机。这种通信方法需要配置连接（连接表）。该连接在一个站的全启动时建立并且一直保持。在 CPU 上需要有自由的连接。

6.2.2 S7-200 PLC与S7-300 PLC间的MPI通信

S7-200 PLC 与 S7-300 PLC 间的 MPI 通信只能采用单边无组态通信，也就是通信无需组态。以下用一个例子说明这种通信的方法。

【例 6-1】 有两台设备，分别由一台 CPU 314C-2DP 和一台 CPU 226CN 控制，从设备 1 上的 CPU 314C-2DP 发出起/停控制命令，设备 2 的 CPU 226CN 收到命令后，对设备 2 进行起停控制，同时设备 1 上的 CPU 314C-2DP 监控设备 2 的运行状态。

【解】

将设备 1 上的 CPU 314C-2DP 作为主站，主站的 MPI 地址为 2，将设备 2 上的 CPU 226CN 作为从站，从站的 MPI 地址为 3。

1．主要软硬件配置

① 1 套 STEP 7 V5.4 SP4。

② 1 台 CPU 314C-2DP。

③ 1 台 CPU 226CN。

④ 1 台 EM277。

⑤ 1 根编程电缆（或者 CP5611 卡）。

⑥ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

⑦ 1 套 STEP 7 Micro/WIN V4.0。

MPI 通信硬件配置图如图 6-3 所示，PLC 接线图如图 6-4 所示。

从图 6-3 可以看出 S7-200 PLC 与 S7-300 PLC 间的 MPI 通信有两种配置方案。方案 1 只要将 PROFIBUS 网络电缆（含两个网络总线连接器）连接在 S7-300 PLC 的 MPI 接口和 S7-200 PLC 的 PPI 接口上即可，而方案 2 却需要另加一个 EM277 模块，显然成本多一些，

但若 S7-200 PLC 的 PPI 接口不够用时，方案 2 是可以选择的配置方案。

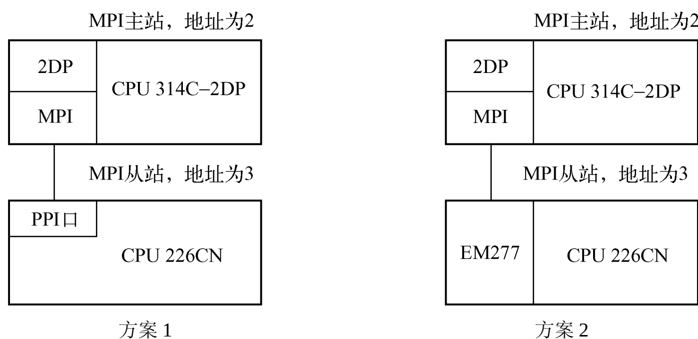


图 6-3　MPI 通信硬件配置图

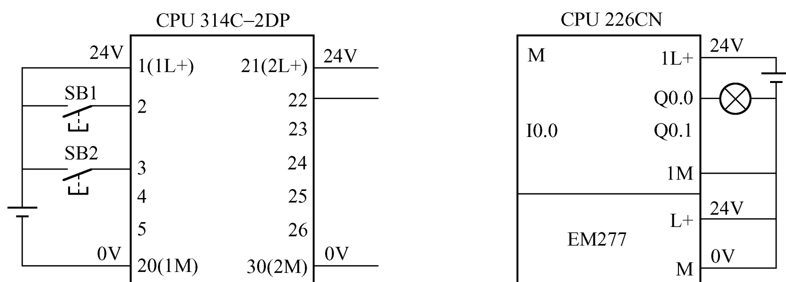


图 6-4　PLC 接线图

2．硬件组态

S7-200 PLC 与 S7-300 PLC 间的 MPI 通信只能采用无组态通信，无组态通信指通信无须组态，完成通信任务，只需要编写程序即可。只要用到 S7-300 PLC，硬件组态还是不可缺少的，这点读者必须清楚。

1）新建工程并插入站点。新建工程，命名为“6-1”，再插入站点，重命名为“Master”，如图 6-5 所示，双击“硬件”，打开硬件组态界面。

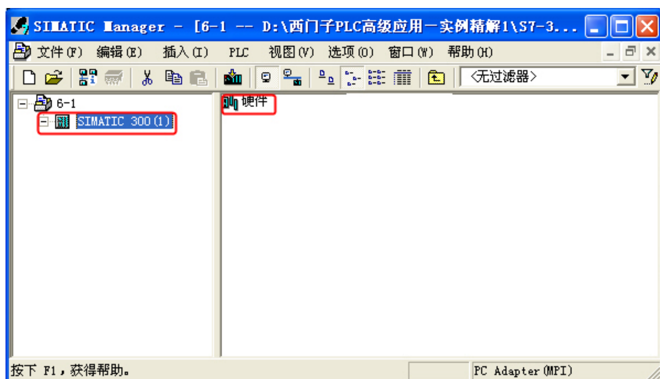


图 6-5　新建工程并插入站点



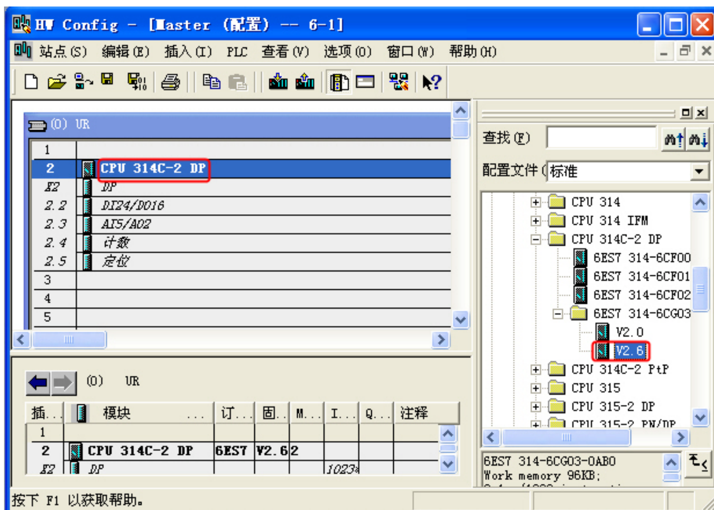


图 6-6　组态主站硬件

2）组态主站硬件。先插入导轨，再插入 CPU 模块，如图 6-6 所示，双击“CPU 314C-2DP”，打开 MPI 通信参数设置界面，单击“属性”按钮，如图 6-7 所示。

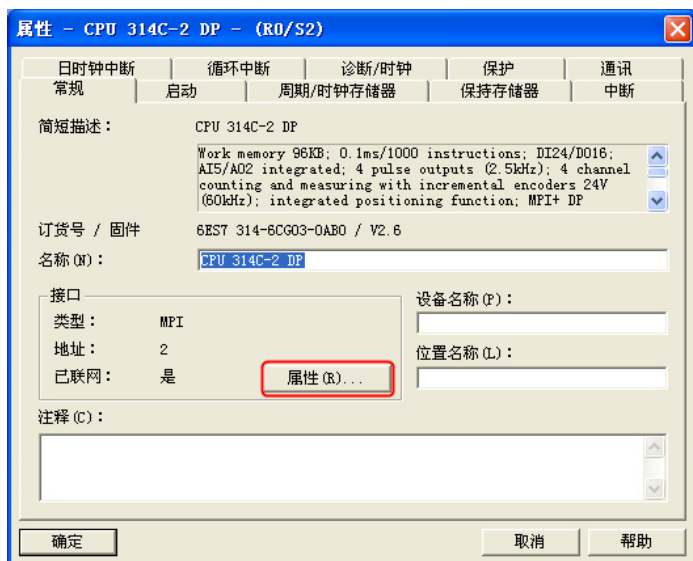


图 6-7　打开 MPI 通信参数设置界面

（3）设置主站的 MPI 通信参数。先选定 MPI 的通信波特率为默认的“187.5Kbps”，再选定主站的 MPI 地址为“2”，再单击“确定”按钮，如图 6-8 所示。最后编译保存和下载硬件组态，在此不再重复叙述。

4）打开系统块。完成以上步骤后，S7-300 PLC 的硬件组态完成，但还必须设置 S7-200 PLC 的通信参数。先打开 STEP 7-Micro/WIN，选定工具条中的“系统块”按钮，并双击之，如图 6-9 所示。

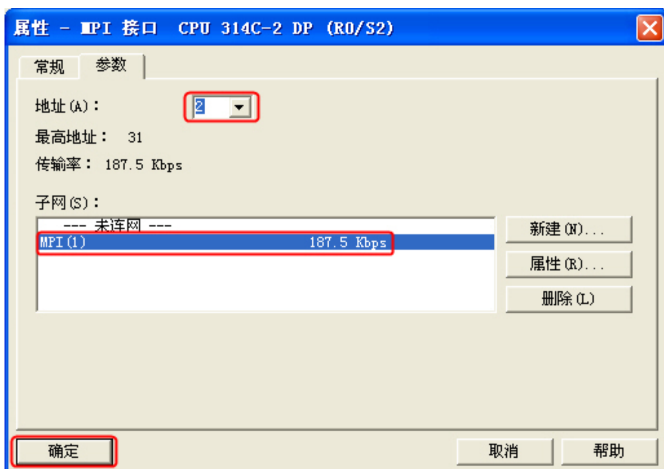


图 6-8　设置主站的 MPI 通信参数

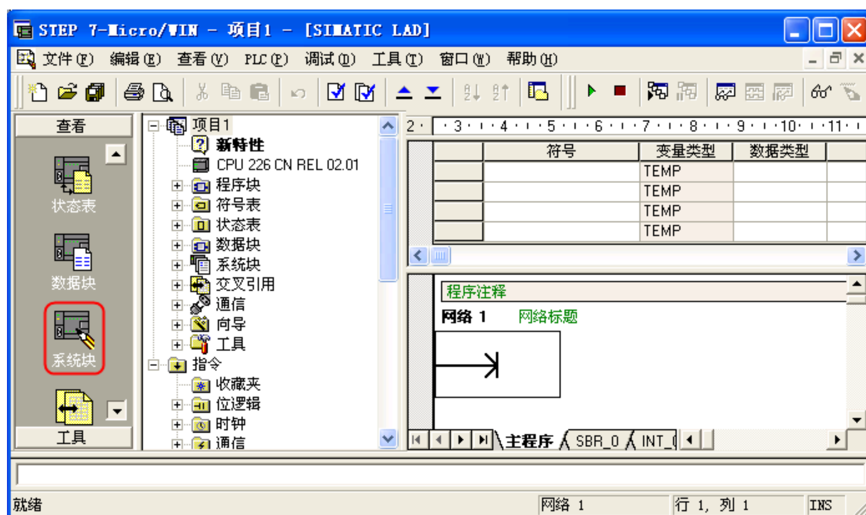


图 6-9　打开系统块

5）设置从站的 MPI 通信参数。先将用于 MPI 通信的接口（本例为 port0）的地址设置成“3”，一定不能设定为“2”，再将波特率设定为“187.5Kbps”，这个数值与 S7-300 PLC 的波特率必须相等，最后单击“确认”按钮，如图 6-10 所示，这一步不少初学者容易忽略，其实这一步非常关键，因为各站的波特率必须相等，这是一个基本原则。系统块设置完成后，还要将其下载到 S7-200 PLC 中，否则通信是不能建立的。

【关键点】硬件组态时，必须将 S7-200 PLC 和 S7-300 PLC 的波特率设置值应相等，此外 S7-300 PLC 的硬件组态和 S7-200 PLC 的系统块必须下载到相应的 PLC 中才能起作用。

3．相关指令介绍

无组态连接的 MPI 的通信适合 S7-400 PLC、S7-300 PLC、S7-200 PLC 之间的通信，通过调用 SFC66、SFC67、SFC68 和 SFC69 来实现。顾名思义，MPI 无组态连接就是 MPI 通信时，不需要组态通信，只要编写通信程序即可实现通信。无组态连接的 MPI 通信分为双



边编程通信方式和单边编程通信方式。S7-200 PLC 与 S7-300 PLC 间的 MPI 通信只能采用单边无组态通信方式。

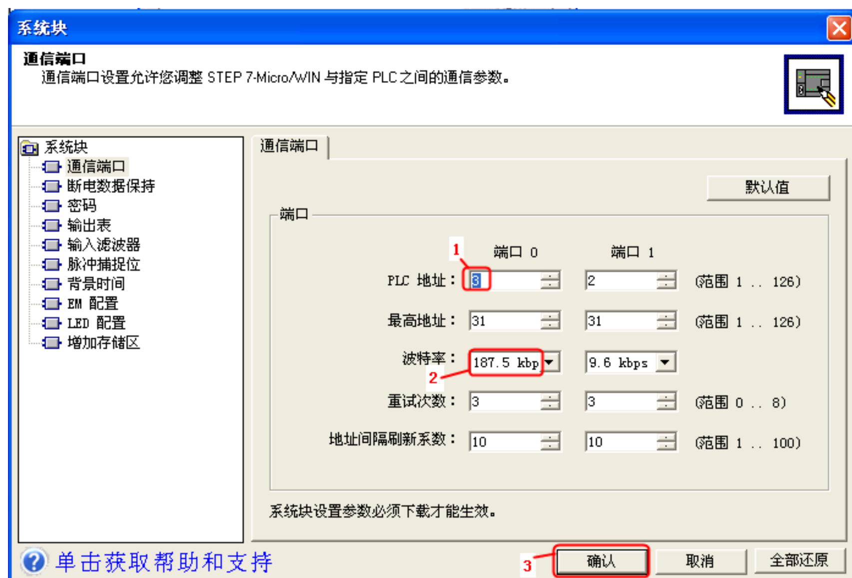


图 6-10　设置从站的 MPI 通信参数

X_PUT（SFC68）发送数据的指令，通过 SFC68 “X_PUT”，将数据写入不在同一个本地 S7 站中的通信伙伴。在通信伙伴上没有相应 SFC。在通过 REQ=1 调用 SFC 之后，激活写作业。此后，可以继续调用 SFC，直到 BUSY=0 指示接收到应答为止。

必须要确保由 SD 参数（在发送 CPU 上）定义的发送区和由 VAR_ADDR 参数（在通信伙伴上）定义的接收区长度相同。SD 的数据类型还必须和 VAR_ADDR 的数据类型相匹配。其输入和输出的含义见表 6-2。

表 6-2　X_PUT（SFC68）指令格式

LAD	输入 / 输出	含　　义	数 据 类 型
"X_PUT" EN　ENO REQ　RET_VAL CONT　BUSY DEST_ID VAR_ADDR SD	EN	使能	BOOL
	REQ	发送请求	BOOL
	CONT	发送完整性	BOOL
	DEST_ID	对方的 MPI 地址	WORD
	VAR_ADDR	对方的数据区	ANY
	SD	本机的数据区	ANY
	RET_VAL	返回数值（如错误值）	INT
	BUSY	发送是否完成	BOOL

X_GET（SFC67）接收数据的指令，通过 SFC67"X_GET"，可以从本地 S7 站以外的通信伙伴中读取数据。在通信伙伴上没有相应 SFC。在通过 REQ=1 调用 SFC 之后，激活读作业。此后，可以继续调用 SFC，直到 BUSY=0 指示数据接收为止。 然后，RET_VAL 便包

含了以字节为单位的、已接收的数据块的长度。

必须要确保由 RD 参数定义的接收区（在接收 CPU 上）至少和由 VAR_ADDR 参数定义的要读取的区域（在通信伙伴上）一样大。RD 的数据类型还必须和 VAR_ADDR 的数据类型相匹配。其输入和输出的含义见表 6-3。

表 6-3　X_GET（SFC67）指令格式

LAD	输入 / 输出	含　义	数据类型
"X_GET" EN　ENO REQ　RET_VAL CONT　BUSY DEST_ID　RD VAR_ADDR	EN	使能	BOOL
	REQ	接收请求	BOOL
	CONT	接收完整性	BOOL
	DEST_ID	对方的 MPI 地址	WORD
	VAR_ADDR	对方的数据区	ANY
	RD	本机的数据区	ANY
	RET_VAL	返回数值（如错误值）	INT
	BUSY	接收是否完成	BOOL

4．程序编写

X_PUT（SFC68）发送数据的指令和 X_GET（SFC67）接收数据的指令是系统功能，也就是系统预先定义的功能，只要将“库”展开，再展开“Standart libarary（标准库）”，选定“X_PUT”或者“X_GET”，再双击之，“X_PUT”或者“X_GET”就自动在网络中指定的位置弹出，如图 6-11 所示。

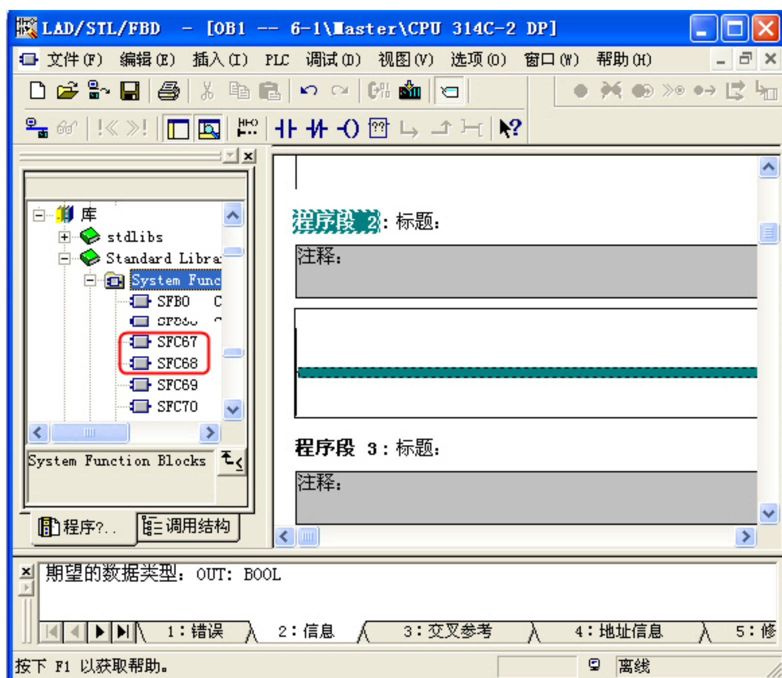
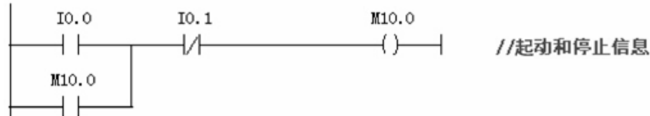


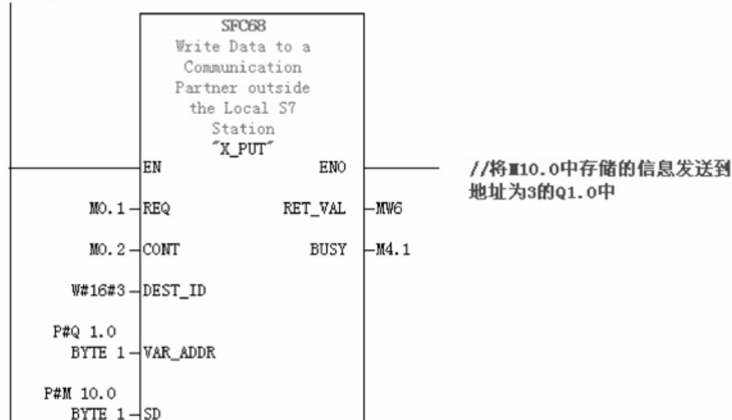
图 6-11　X_PUT 和 X_GET 指令的位置



主站的程序如图 6-12 所示，从站并不需要编写程序。







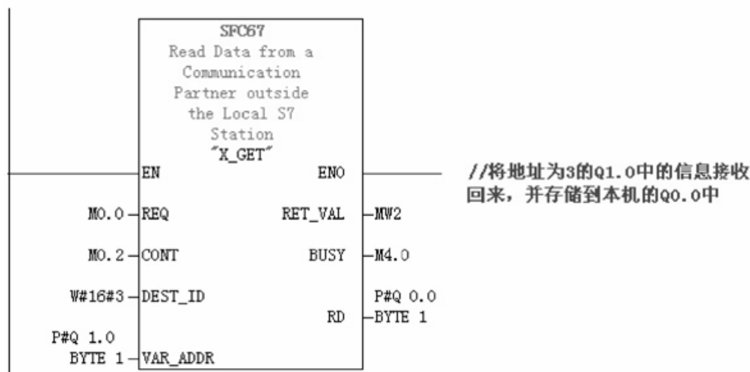


图 6-12　主站程序

【关键点】本例主站地址为“2”，从站的地址为“3”，因此硬件配置采用方案 1 时，必须将“PPI 口”的地址设定为“3”。而采用方案 2 时，必须将 EM277 的地址设定为“3”，设定完成后，还要将 EM277 断电，新设定的地址才能起作用。指令“X_PUT”的参数 SD 和 VAR_ADDR 的数据类型可以据实际情况确定，但在同一程序中数据类型必须一致。

【例 6-2】 某设备上有一台 CPU 226CN，CPU 226CN 上的两个 PPI 接口已经被占用，现要改造此设备，需要添加一台 TP177B 触摸屏，请设计一个方案。

【解】

CPU 226CN 上的两个 PPI 接口已经被占用，所以要建立 TP177B 触摸屏和 CPU 226CN 的通信，就必须扩展 CPU 226CN 通信接口。扩展一个 EM277 模块即可，TP177B 触摸屏和 CPU 226CN 可以采用 MPI 或者 PROFIBUS 通信，硬件配置方案如图 6−13 所示。



图 6−13　硬件配置方案

6.2.3　S7-300 PLC与S7-300 PLC间的MPI通信

S7-300 PLC 与 S7-300 PLC 间的 MPI 通信除了可以采用前述的无组态通信方式外，还可以采用全局数据通信方式，这种通信方式可以在 S7-300 PLC 与 S7-300 PLC、S7-400 PLC 与 S7-300 PLC、S7-400 PLC 与 S7-400 PLC 之间通信，用户不需要编写程序，在硬件组态时，组态所有 MPI 的 PLC 站之间的发送区与接收区即可。以下用一个例子介绍 S7-300 PLC 与 S7-300 PLC 之间的全局数据 MPI 通信。

【例 6−3】 有两台设备，各由一台 CPU 314C-2DP 控制，从设备 1 上的 CPU 314C-2DP 发出起/停控制命令，设备 2 的 CPU 314C-2DP 收到命令后，对设备 2 进行起/停控制，同时设备 1 上的 CPU 314C-2DP 监控设备 2 的运行状态。

【解】

将设备 1 上的 CPU 314C-2DP 作为主站，主站地址为 2，将设备 2 上的 CPU 314C-2DP 作为从站，从站地址为 3。

1．主要软硬件配置

① 1 套 STEP 7 V5.4 SP4。

② 2 台 CPU 314C-2DP。

③ 1 根编程电缆（或者 CP5611 卡）。

④ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

MPI 通信硬件配置图如图 6−14 所示，PLC 接线图如图 6−15 所示。

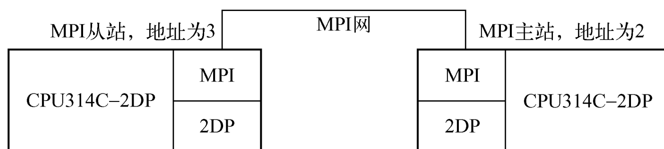


图 6−14　MPI通信硬件配置图

2．硬件组态

1）新建工程和插入站点。新建工程，本例的工程名为“6-2”，再在工程中插入两个站点 SIMATIC 300（2）和 SIMATIC 300（3），选定站点“SIMATIC 300（2）”，双击“Hardware”，如图 6−16 所示。



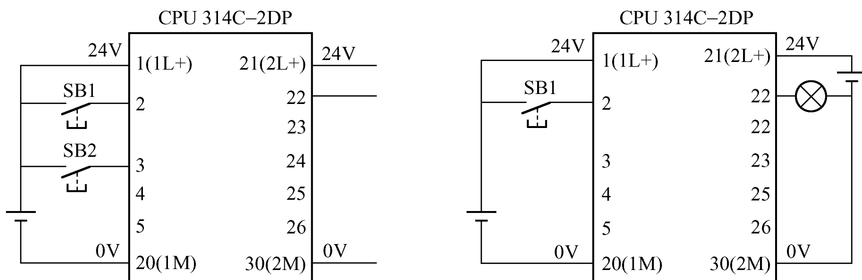


图 6–15　MPI通信PLC接线图

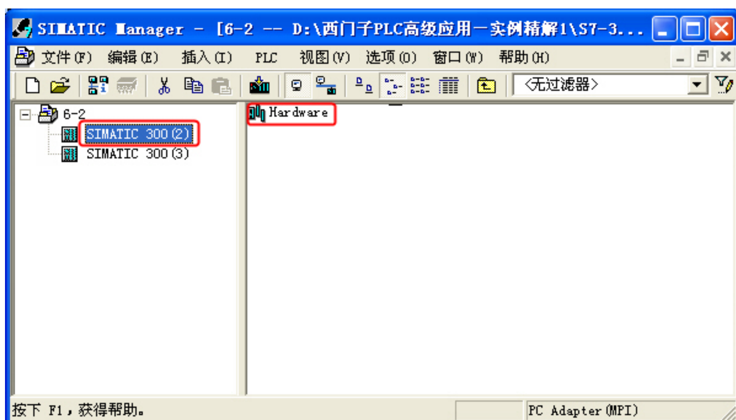


图 6–16　新建工程和插入站点

2）插入导轨。双击“Rail”，弹出导轨，如图 6–17 所示。

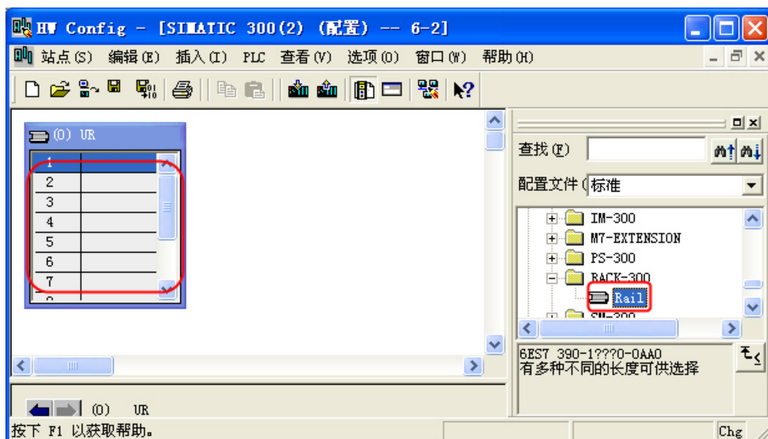


图 6–17　插入导轨

3）打开 CPU 314C-2DP 属性。双击槽位 2 的“CPU 314C-2DP”，如图 6–18 所示。

4）设置站 2 的 MPI 通信参数。单击“属性”按钮，如图 6–19 所示，弹出设置 MPI 通信参数界面，如图 6–20 所示，设定 MPI 的地址为“2”，MPI 的通信波特率为“187.5Kbps”，再单击“确定”按钮。





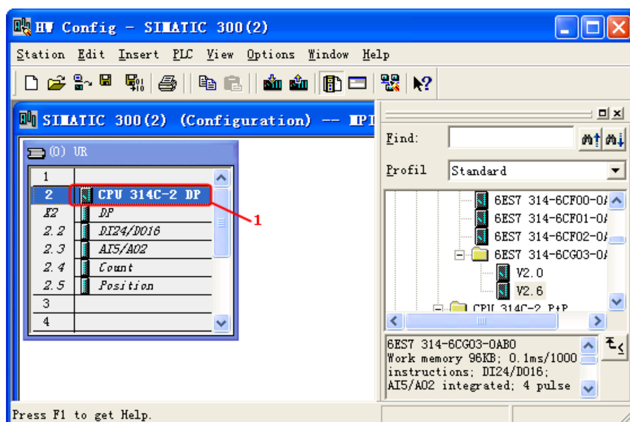


图 6-18　打开 CPU 314C-2DP 属性

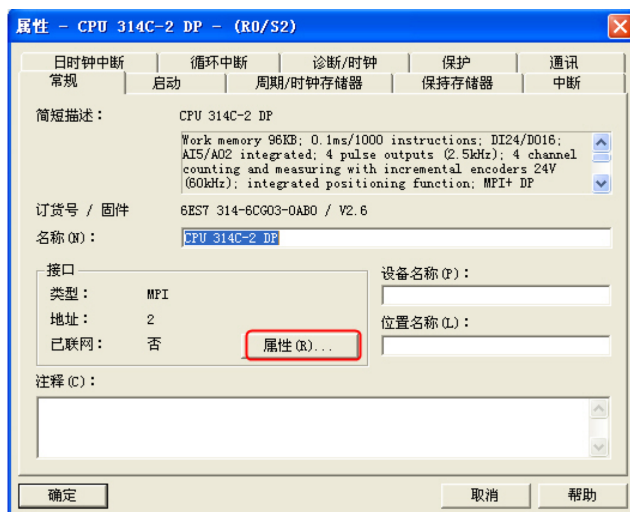


图 6-19　打开 MPI 通信参数设置界面

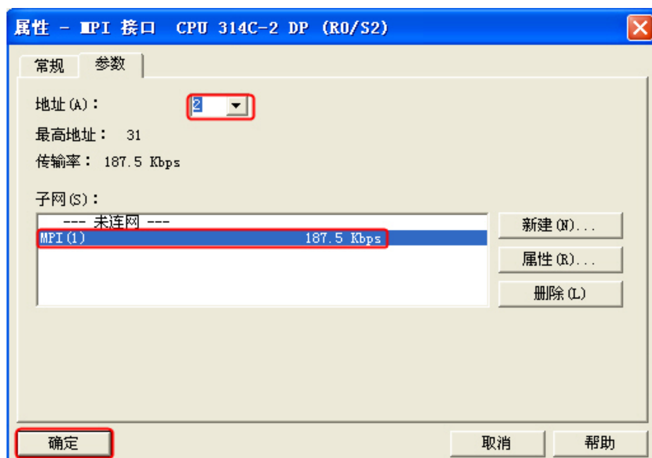


图 6-20　设置站 2 的 MPI 通信参数



5）站 3 的硬件组态。回到图 6-16，选定“SIMATIC 300（3）”，双击“Hardware”，弹出硬件组态界面，先插入导轨，再插入 CPU 模块，如图 6-21 所示。双击槽位 2 的“CPU 314C-2DP”。

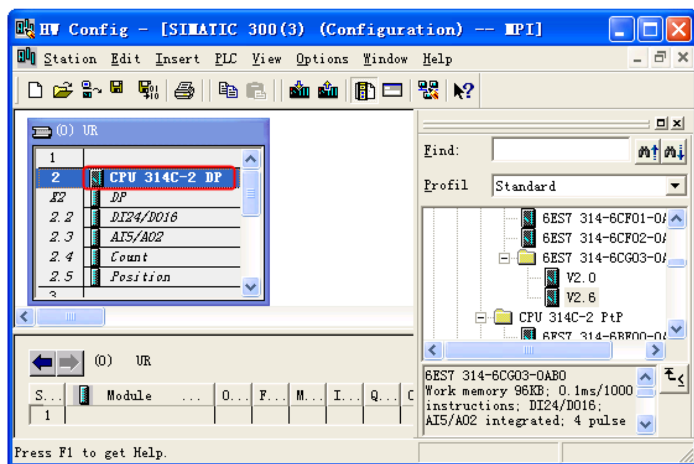


图 6-21　站 3 的硬件组态

6）打开 MPI 通信参数设置界面。单击“属性”按钮，如图 6-22 所示，弹出站 3 的 MPI 通信参数设置界面，如图 6-23 所示，设定 MPI 的地址为“3”，MPI 的通信波特率为“187.5Kbps”，再单击“确定”按钮。

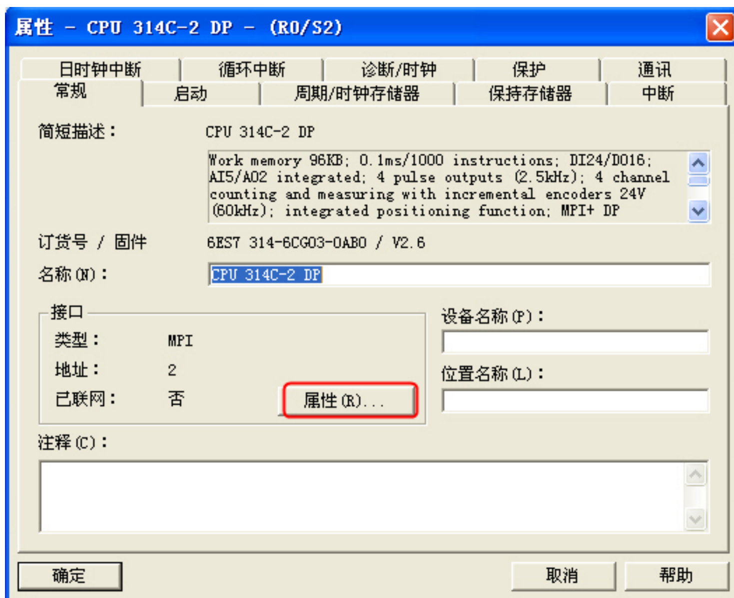


图 6-22　打开 MPI 通信参数设置界面

7）打开 MPI 网络。双击标记“1”处的“MPI（1）”，如图 6-24 所示，弹出 MPI 的网络，如图 6-25 所示。

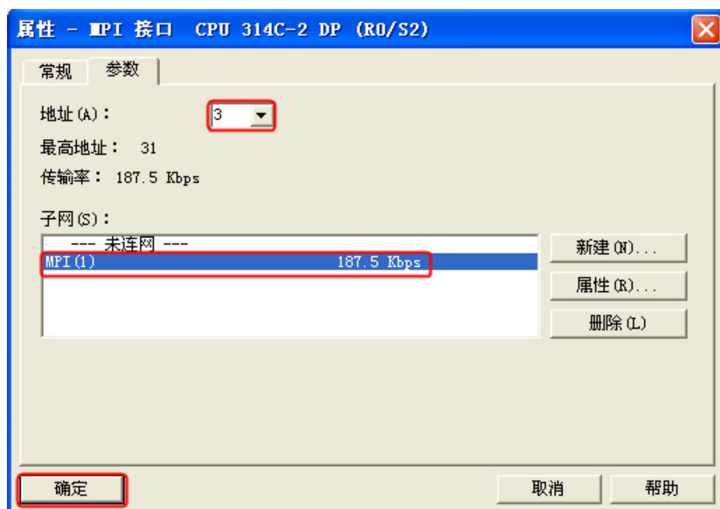


图 6-23　设置站 3 的 MPI 通信参数

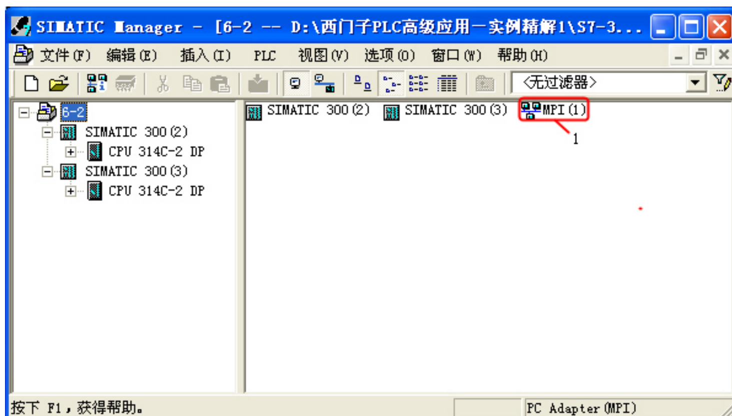


图 6-24　打开 MPI 网络（1）

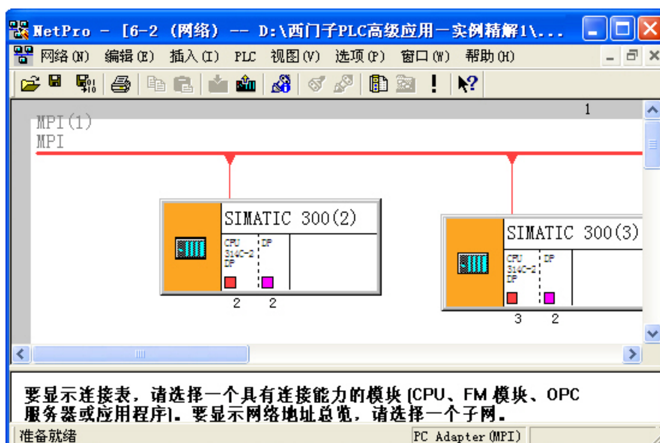


图 6-25　打开 MPI 网络（2）



8）打开全局变量发送、接收区组态。选中标记“1”处的“MPI（1）”网络线，再选中菜单“选项”，单击子菜单“定义全局数据”，打开全局变量发送、接收区组态如图 6-26 所示。

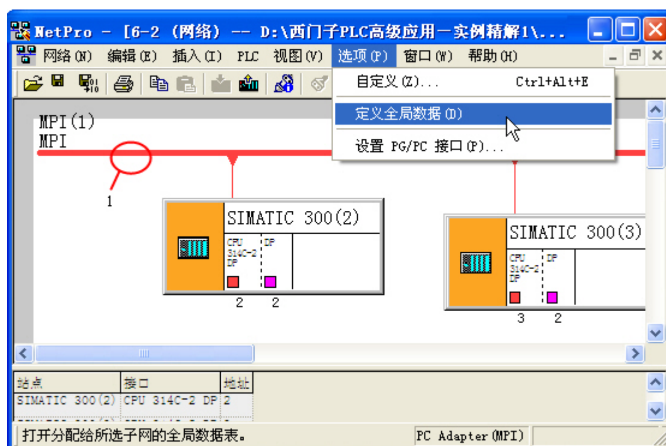


图 6-26　打开全局变量发送、接收区组态

9）MPI 全局变量组态。双击标记“1”处，如图 6-27 所示。



图 6-27　MPI 全局变量组态

10）选定 CPU。选定“SIMATIC（2）”，再选定“CPU 314C-2DP”，再单击“确定”按钮，如图 6-28 所示。

11）定义发送区的数据组。输入 MB10:5，其含义是：将站点 SIMATIC 300（2） MB10 开始的 5 个字节发送出去，如图 6-29 所示。

12）发送区的数据组的组态。选定“编辑”菜单，单击“发送器”，定义发送区的数据组，如图 6-30 所示，其他发送区和接收的数据组的组态方法类似，如图 6-31 所示。含义

是：将站点 SIMATIC 300（2）的从 MB10 开始的 5 个字节发送到 SIMATIC 300（3）的从 MB10 开始的 5 个字节的存储区中，将站点 SIMATIC 300（3）的从 MB30 开始的 5 个字节发送到 SIMATIC 300（2）的从 MB30 开始的 5 个字节的存储区中。具体数据流向见表 6-4。

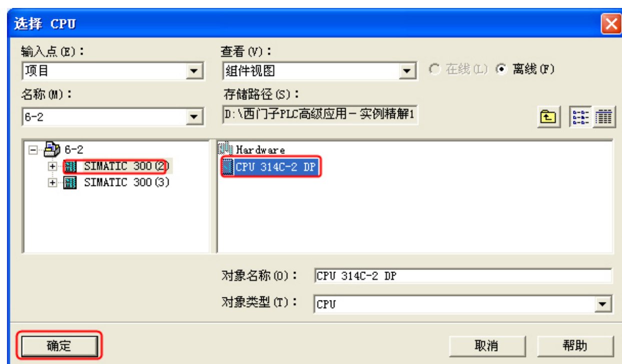


图 6-28　选定 CPU



图 6-29　定义发送区的数据组



图 6-30　发送区的数据组的组态



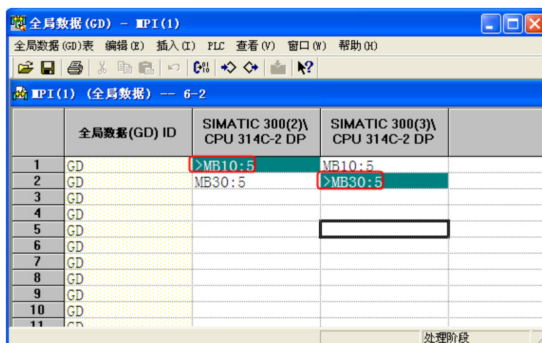


图 6-31　发送区和接收的数据组的组态

表 6-4　全局 MPI 数据流向

序号	SIMATIC 300（2）	对应关系	SIMATIC 300（3）
1	MB10～MB14	⟶	MB10～MB14
2	MB30～MB34	⟵	MB30～MB34

13）编译和保存组态内容。单击“保存”按钮即可，如图 6-32 所示。

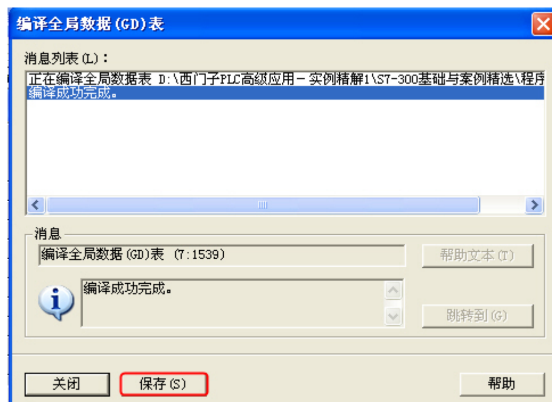


图 6-32　保存组态内容

14）下载组态信息。单击工具栏中的“下载”按钮，如图 6-33 所示。选定 SIMATIC 300（2）和 SIMATIC 300（3）分别下载到对应的站点中去，如图 6-34 所示。

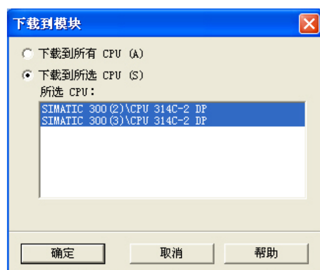


图 6-33　下载组态信息（1）

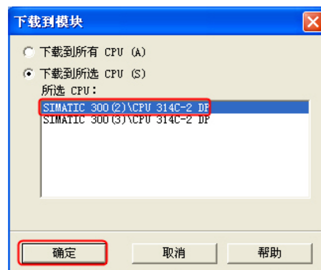


图 6-34　下载组态信息（2）





15）组态完成。组态完成后，经过编译后，界面如图 6-35 所示。GD X.Y.Z（如 GD 1.2.1）的含义见表 6-5。

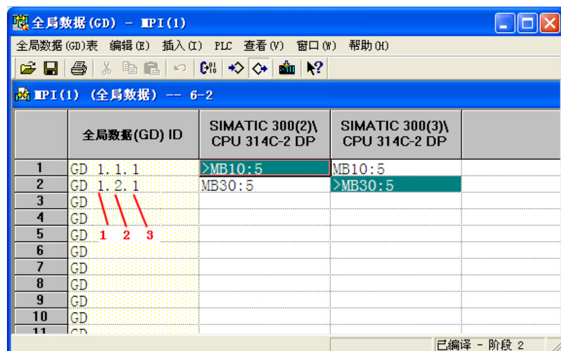


图 6-35　组态完成界面

表 6-5　GD X.Y.Z 的含义

序号	参数	含义
1	X	全局变量数据包的循环次数，次数与 CPU 有关，S7-300 最多支持 4 个
2	Y	一个循环中有几个数据包
3	Z	是一个数据包中的数据区

3．编写程序

全局数据的 MPI 通信，只要硬件进行组态就可以通信了，通信部分是不要编写程序的，像本例这样简单的工程，若组态合理不需要编写一条程序。但一个实际的工程不编写程序是很不现实的。主站 2 和从站 3 的程序如图 6-36 和图 6-37 所示。

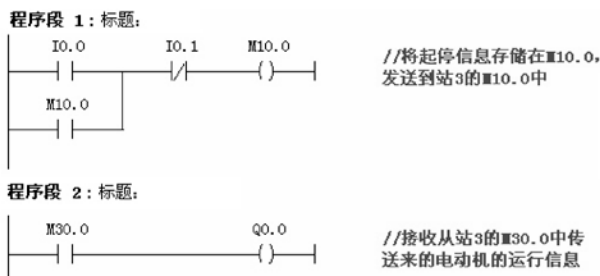


图 6-36　主站 2 的程序

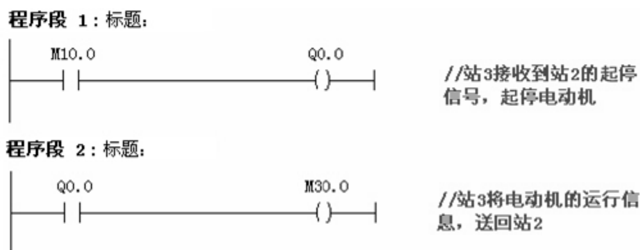


图 6-37　从站 3 的程序



【关键点】本例的关键在于将 MPI 的通信组态正确，还有一点要特别注意，就是站 2 哪个数据区将数据送到站 3 哪个数据区中，站 2 又从站 3 哪个数据区接收数，这些关系是绝对不能弄错的，否则不可能建立正确的通信。

6.2.4　S7-300/400 PLC与S7-400 PLC间的MPI通信

在前面的章节讲解了无组态 MPI 通信、全局数据 MPI 通信，这些 MPI 的通信方式其实都适用于 S7-300/400 PLC 与 S7-400 PLC 间的 MPI 通信，但组态 MPI 通信方式就只适用于后者，以下用一个例子讲解 S7-300 PLC 与 S7-400 PLC 间的 MPI 通信。组态方式的 MPI 通信的好处是处理的数据量大。

【例 6−4】 有两台设备，分别由一台 CPU 314C-2DP 和一台 CPU 416-2DP 控制，从设备 1 上的 CPU 416-2DP 发出起停控制命令，设备 2 的 CPU 314C-2DP 收到命令后，对设备 2 进行起停控制，同时设备 1 上的 CPU 314C-2DP 监控设备 2 的运行状态。

【解】

将设备 1 上的 CPU 416-2DP 作为主站，主站地址为 2，将设备 2 上的 CPU 314C-2DP 作为从站，从站地址为 3。

1．主要软硬件配置

① 1 套 STEP 7 V5.4 SP4。

② 1 台 CPU 314C-2DP。

③ 1 台 CPU 416-2DP。

④ 1 根编程电缆（或者 CP5611 卡）。

⑤ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

MPI 通信硬件配置图如图 6−38 所示。

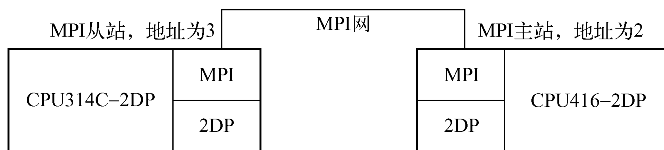


图 6−38　MPI 通信硬件配置图

2．硬件组态

1）新建工程。新建工程，命名为“6−4”，插入站点和 CPU，并将建立 CPU 314C-2DP 和 CPU 416-2DP 的 MPI 连接，其中 CPU 416-2DP 的 MPI 地址为“2”，CPU 314C-2DP 的 MPI 地址为“3”，如图 6−39 所示，再单击“MPI（1）”标志，弹出如图 6−40 所示的界面。

2）新建连接。如图 6−40 所示，选中“1”处，单击右键，弹出快捷菜单，单击“插入新连接”，弹出如图 6−41 所示的界面。

3）选择 CPU 的连接方式。如图 6−41 所示，选中“CPU 314C-2DP”和“S7 连接”，单击“应用” 按钮，弹出如图 6−42 所示的界面。

4）选择 MPI 参数。如图 6−42 所示，，单击“确定”按钮，硬件组态完成。





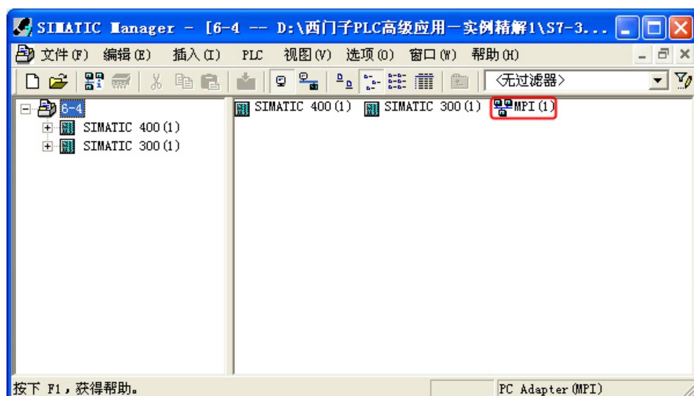


图 6-39　新建工程

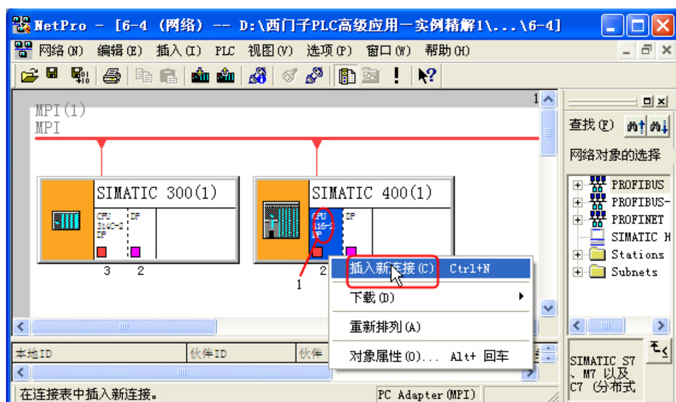


图 6-40　新建连接

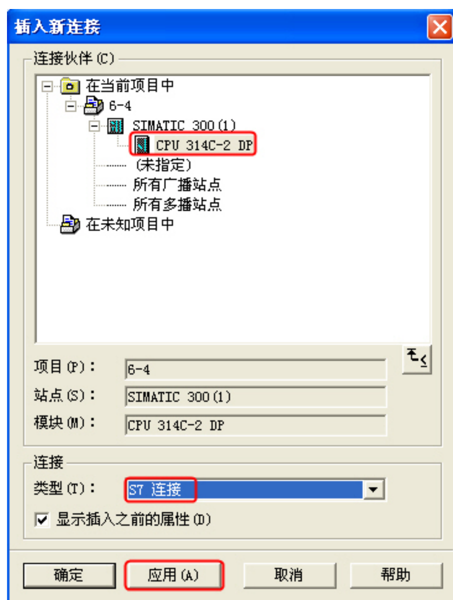


图 6-41　选择 CPU 的连接方式



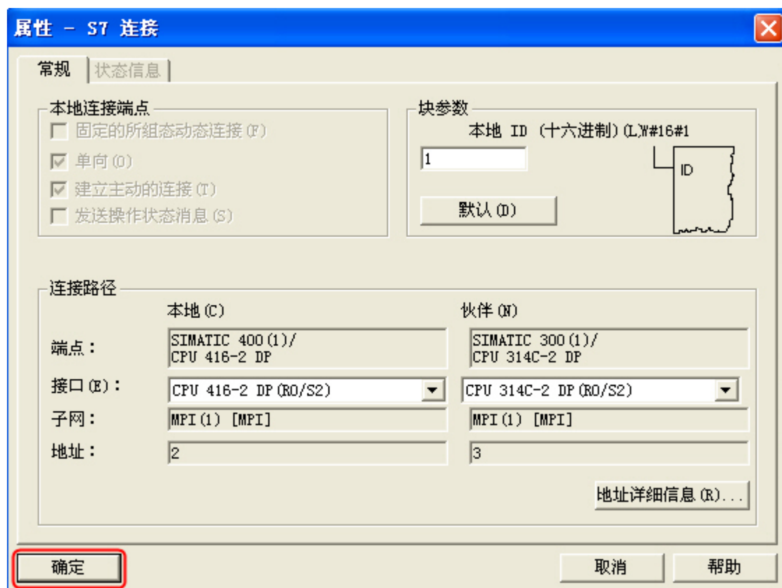


图 6-42 选择 MPI 参数

3. 相关指令介绍

PUT（SFB15）是发送指令，通过使用 SFB15“PUT”，可以将数据写入到远程 CPU。对于 S7-400 PLC，在 REQ 的上升沿处发送数据。在 REQ 的每个上升沿处传送参数 ID、ADDR_1 和 SD_1。在每个作业结束之后，可以给 ID、ADDR_1 和 SD_1 参数分配新数值。其各参数的含义见表 6-6。

表 6-6 PUT（SFB15）指令格式

LAD	输入 / 输出	含　义	数据类型
"PUT" EN ENO REQ DONE ID ERROR ADDR_1 STATUS ADDR_2 ADDR_3 ADDR_4 SD_1 SD_2 SD_3 SD_4	EN	使能	BOOL
	REQ	发送请求	BOOL
	ID	地址参数	WORD
	ADDR_1	本地的存储地址	ANY
	SD_1	对方的数据区	ANY
	DONE	是否发送完成	BOOL
	ERROR	是否错误	BOOL
	STATUS	状态	WORD

GET（SFB14）是接收指令，通过 SFB14“GET”，从远程 CPU 中读取数据。对于 S7-400 PLC，在 REQ 的上升沿处读取数据。在 REQ 的每个上升沿处传送参数 ID、ADDR_1 和 RD_1。在每个作业结束之后，可以分配新数值给 ID、ADDR_1 和 RD_1 参数。其各参数的含义见表 6-7。

表 6-7　GET（SFB14）指令格式

LAD	输入 / 输出	含　　义	数据类型
"GET" EN ENO REQ NDR ID ERROR ADDR_1 STATUS ADDR_2 ADDR_3 ADDR_4 RD_1 RD_2 RD_3 RD_4	EN	使能	BOOL
	REQ	接收请求	BOOL
	ID	地址参数	WORD
	ADDR_1	本地的存储地址	ANY
	RD_1	对方的数据区	ANY
	NDR	是否在接收完成	BOOL
	ERROR	是否错误	BOOL
	STATUS	状态	WORD

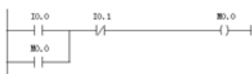
【关键点】PUT（SFB15）和 GET（SFB14）指令的参数 ID 设定如图 6-42 所示，本通信实用 OSI 模型的第 1、2 和 7 层。

4．编写程序

主站的程序如图 6-43 所示，从站的程序如图 6-44 所示。

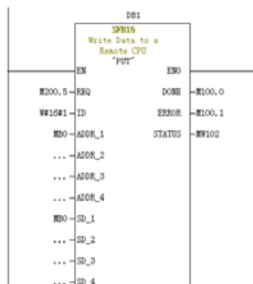
程序段 1：标题：

启停信息



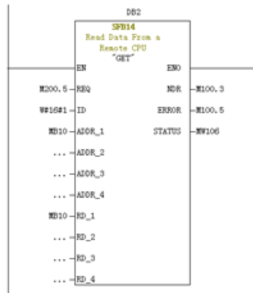
程序段 2：标题：

当M200.5有效时，主站2的MB0向从站3的MB0数据区发送数据



程序段 3：标题：

当M200.5有效时，主站2的MB10接收来自从站3的MB10数据区的数据



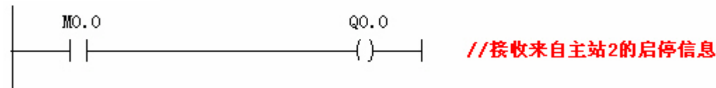
程序段 4：标题：

注释：



图 6-43　主站程序





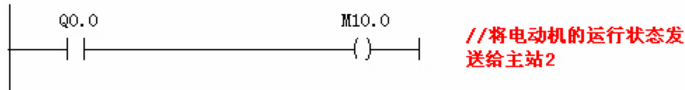


图 6-44　从站程序

6.3　PROFIBUS通信及其应用

6.3.1　PROFIBUS通信基础

1．现场总线及其国际标准

IEC（国际电工委员会）对现场总线（Fieldbus）的定义是："安装在制造和过程区域的现场装置与控制室内的自动控制装置之间的数字式、串行、多点通信的数据总线称为现场总线"。IEC 61158 是迄今为止制订时间最长、意见分歧最大的国际标准之一。制订时间超过 12 年，先后经过 9 次投票，在 1999 年底获得通过。IEC 61158 最后容纳了 8 种互不兼容的协议：

类型 1：原 IEC61158 技术报告，即现场总线基金会（FF）的 H1。

类型 2：Control Net（美国 Rockwell 公司支持）。

类型 3：PROFIBUS（德国西门子公司支持）。

类型 4：P-Net（丹麦 Process Data 公司支持）。

类型 5：FF 的 HSE（原 FF 的 H2，高速以太网，美国 Fisher Rosemount 公司支持）。

类型 6：Swift Net（美国波音公司支持）。

类型 7：WorldFIP（法国 Alstom 公司支持）。

类型 8：Interbus（德国 Phoenix contact 公司支持）。

各类型将自己的行规纳入 IEC 61158，且遵循两个原则：

1）不改变 IEC 61158 技术报告的内容。

2）8 种类型都是平等的，类型 2～8 都对类型 1 提供接口，标准并不要求类型 2～8 之间提供接口。

2．工厂自动化网络结构

（1）现场设备层

主要功能是连接现场设备，例如分布式 I/O、传感器、驱动器、执行机构和开关设备等，完成现场设备控制及设备间联锁控制。

（2）车间监控层

车间监控层又称为单元层，用来完成车间主生产设备之间的连接，包括生产设备状态的在线监控、设备故障报警及维护等。还有生产统计、生产调度等功能。传输速度不是最重要的，但是应能传送大容量的信息。

（3）工厂管理层

车间操作员工作站通过集线器与车间办公管理网连接，将车间生产数据送到车间管理层。车间管理网作为工厂主网的一个子网，连接到厂区骨干网，将车间数据集成到工厂管理层。工厂自动化网络结构如图 6-45 所示。

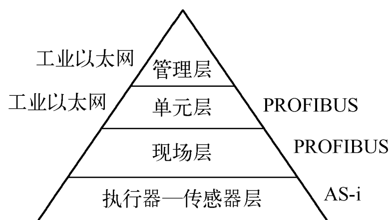


图 6-45　工厂自动化网络结构

3．PROFIBUS 的类型

PROFIBUS 已被纳入现场总线的国际标准 IEC 61158 和欧洲标准 EN50170，并于 2001 年被定为我国的国家标准 JB/T10308.6-2001。PROFIBUS 在 1999 年 12 月通过的 IEC 61156 中称为 Type 3，PROFIBUS 的基本部分称为 PROFIBUS-V0。在 2002 年新版的 IEC61156 中增加了 PROFIBUS-V1，PROFIBUS-V2 和 RS-485IS 等内容。新增的 PROFInet 规范作为 IEC 61158 的 Type10。截止 2003 年年底，安装的 PROFIBUS 节点设备已突破了 1 千万个，在中国超过 150 万个。

ISO/OSI 通信标准由 7 层组成，并分两类。一类是面向用户的第 5 层到第 7 层，一类是面向网络的第 1 到第 4 层。第 1 到第 4 层描述了数据从一个地方传输到另一个地方，第 5 层到第 7 层给用户提供适当的方式访问网络系统。PROFIBUS 协议使用了 ISO/OSI 模型的第 1 层、第 2 层和第 7 层。

从用户的角度看，PROFIBUS 提供 3 种通信协议类型：PROFIBUS-FMS、PROFIBUS-DP 和 PROFIBUS-PA。

1）PROFIBUS-FMS （Fieldbus Message Specification，现场总线报文规范），使用了第 1 层、第 2 层和第 7 层。第 7 层（应用层）包含 FMS 和 LLI（底层接口）主要用于系统级和车间级的不同供应商的自动化系统之间传输数据，处理单元级（PLC 和 PC）的多主站数据通信。

2）PROFIBUS-DP（Decentralized Periphery，分布式外部设备），使用第 1 层和第 2 层，这种精简的结构特别适合数据的高速传送，PROFIBUS-DP 用于自动化系统中单元级控制设备与分布式 I/O（例如 ET 200）的通信。主站之间的通信为令牌方式，主站与从站之间为主从方式，以及这两种方式的混合。

3）PROFIBUS-PA（Process Automation，过程自动化）用于过程自动化的现场传感器和执行器的低速数据传输，使用扩展的 PROFIBUS-DP 协议。传输技术采用 IEC 1158-2 标准，可以用于防爆区域的传感器和执行器与中央控制系统的通信。使用屏蔽双绞线电缆，由总线提供电源。此外，基于 PROFIBUS，还推出了用于运动控制的总线驱动技术 PROFI-drive 和故障安全通信技术 PROFI-safe。

此外，对于西门子系统，PROFIBUS 提供了两种更为优化的通信方式，即 PROFIBUS-S7 通信和 S5 兼容通信。

1）PROFIBUS-S7（PG/OP 通信）使用了第 1 层、第 2 层和第 7 层。特别适合 S7 PLC 与 HMI 和编程器通信，也可以用于 S7-300 PLC 和 S7-400 PLC 以及 S7-400 PLC 和 S7-400 PLC 之间的通信。

2）PROFIBUS-FDL（S5 兼容通信）使用了第 1 层和第 2 层。数据传送快，特别适合 S7-300 PLC、S7-400 PLC 和 S5 PLC 之间的通信。



6.3.2 S7-200 PLC与S7-300 PLC间的现场总线通信

以下以一台 CPU 314C-2DP 与一台 CPU 226CN 之间的 PROFIBUS 的现场总线通信为例介绍 S7-200 PLC 与 S7-300 PLC 间的现场总线通信。

【例 6−5】 模块化生产线的主站为 CPU 314C-2DP，从站为 CPU 226CN 和 EM277 的组合，主站发出开始信号（开始信号为高电平），从站接收信息，并使从站的指示灯以 1s 为周期闪烁。

【解】

1．主要软硬件配置

① 1 套 STEP 7-Micro/WIN V4.0。

② 1 套 STEP 7 V5.4 SP4。

③ 1 台 CPU 226CN。

④ 1 台 EM277。

⑤ 1 台 CPU 314C-2DP。

⑥ 1 根编程电缆（或者 CP5611 卡）。

⑦ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置如图 6−46 所示，PROFIBUS 现场总线通信 PLC 接线如图 6−47 所示。

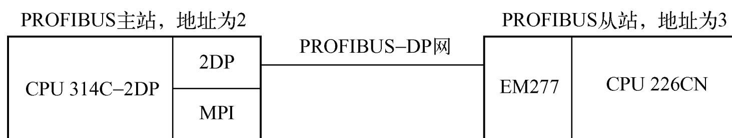


图 6−46 PROFIBUS 现场总线硬件配置图

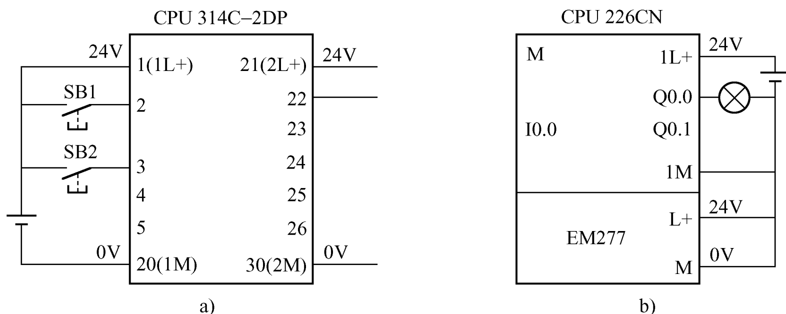


图 6−47 PROFIBUS 现场总线通信 PLC 接线图

a) CPU 314C-2DP PLC 接线图 b) CPU 226CN PLC 接线图

2．CPU 314C-2DP的硬件组态

S7-300 PLC 与 S7-200 PLC 的 PROFIBUS 通信总的方法是：首先对主站 CPU 314C-2DP 的硬件进行硬件组态，下载硬件，再编写主站程序，下载主站程序；编写从站程序，下载从站程序，最后便可建立主站和从站的通信。具体步骤如下：

1）打开 STEP 7 软件。双击桌面上的快捷键，打开 STEP 7 软件，如图 6−48 所示。

当然也可以单击“开始”→“所有程序”→“SIMATIC”→“SIMATIC Manager”打开 STEP 7 软件。

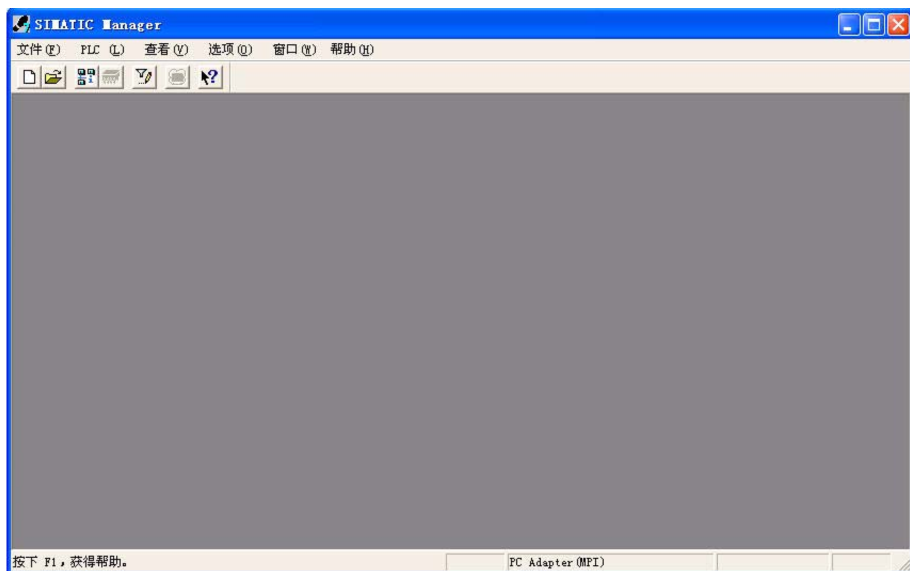


图 6-48　打开 STEP 7 软件

2）新建项目。单击“新建”按钮，弹出“新建 项目”对话框，在“命名（M）”中输入一个名称，本例为“profibus”再单击“确定”按钮，如图 6-49 和图 6-50 所示。

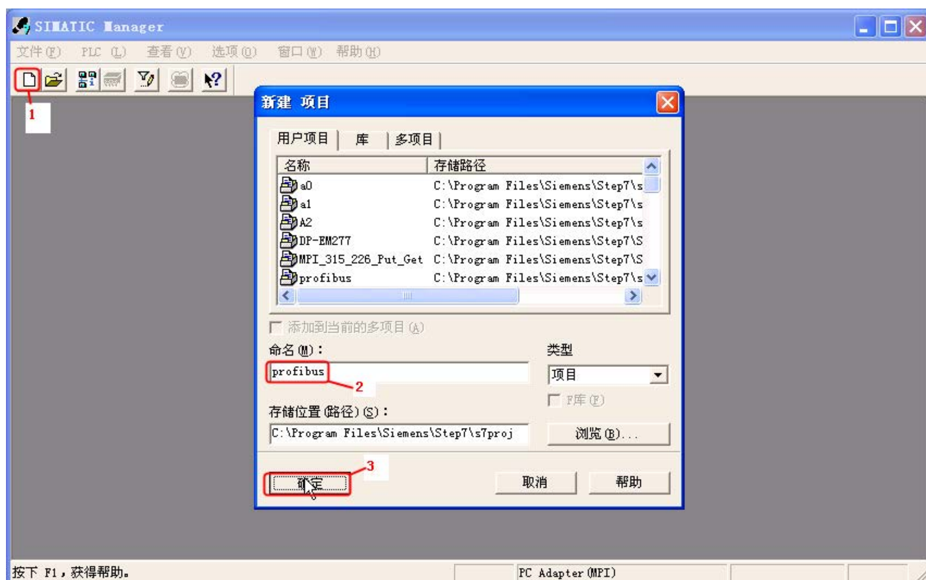


图 6-49　新建项目（1）

3）插入站点。单击菜单栏“插入”菜单，再单击“站点”和“SIMATIC 300 站点”子菜单，如图 6-51 和图 6-52 所示。这个步骤的目的主要是为了插入主站。



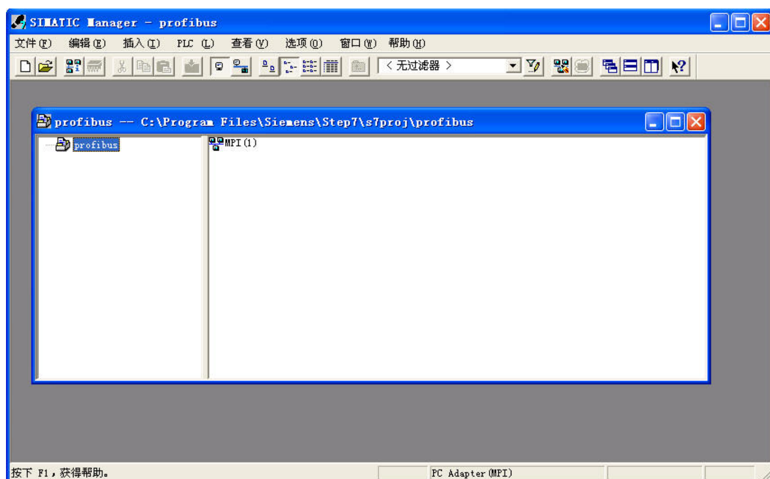


图 6-50　新建项目（2）

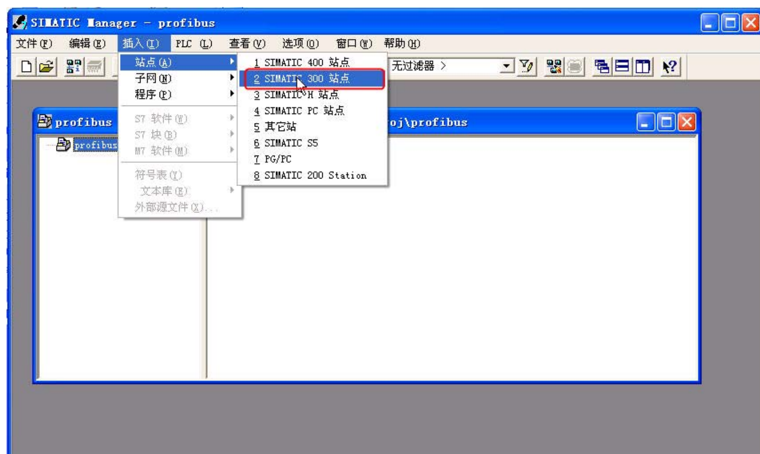


图 6-51　插入站点（1）

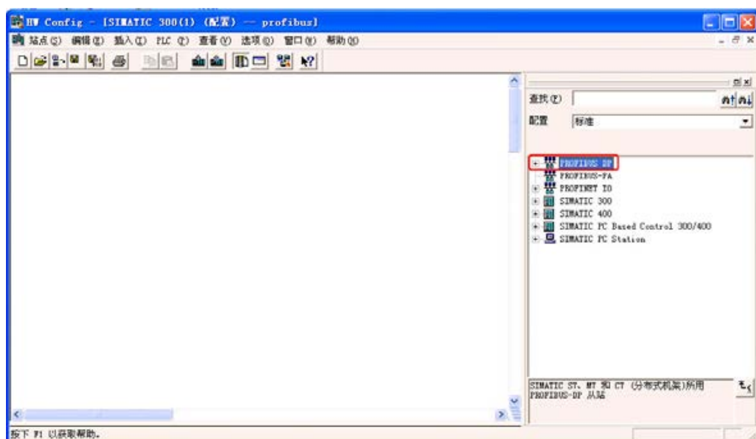


图 6-52　插入站点（2）

4）插入导轨。展开项目中的“SIMATIC 300”下的“RACK-300”，双击导轨“Rail”，如图 6-53 所示。硬件配置的第一步都是加入导轨，否则下面的步骤不能进行。

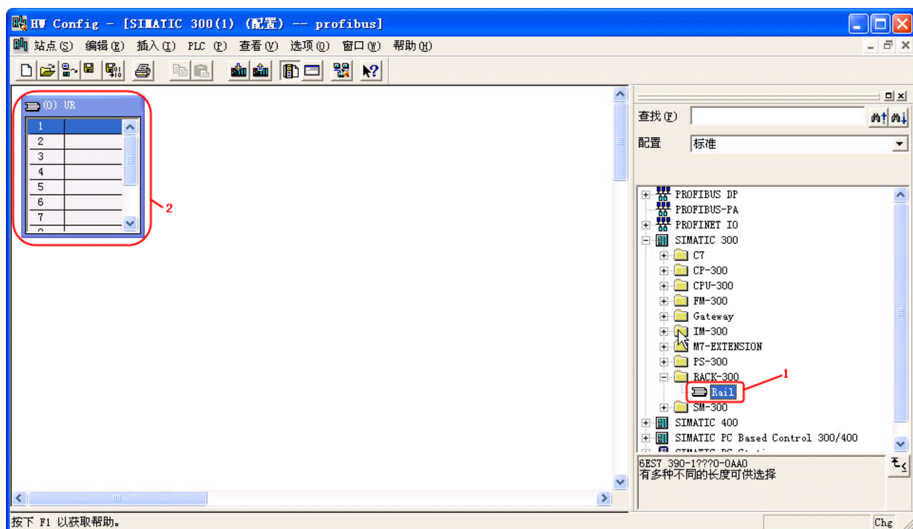


图 6-53　插入导轨

5）插入 CPU。展开项目中的“SIMATIC 300”下的“CPU-300”，再展开“CPU 314C-2DP”下的“6ES7 314-6CG06-OABO”，将“V2.6”拖入导轨的 2 号槽中，如图 6-54 所示。若选用了西门子的电源，在配置硬件时，应该将电源加入到第一槽，本例中使用的是开关电源，因此硬件配置时不需要加入电源，但第一槽必须空缺，建议读者最好选用西门子电源。

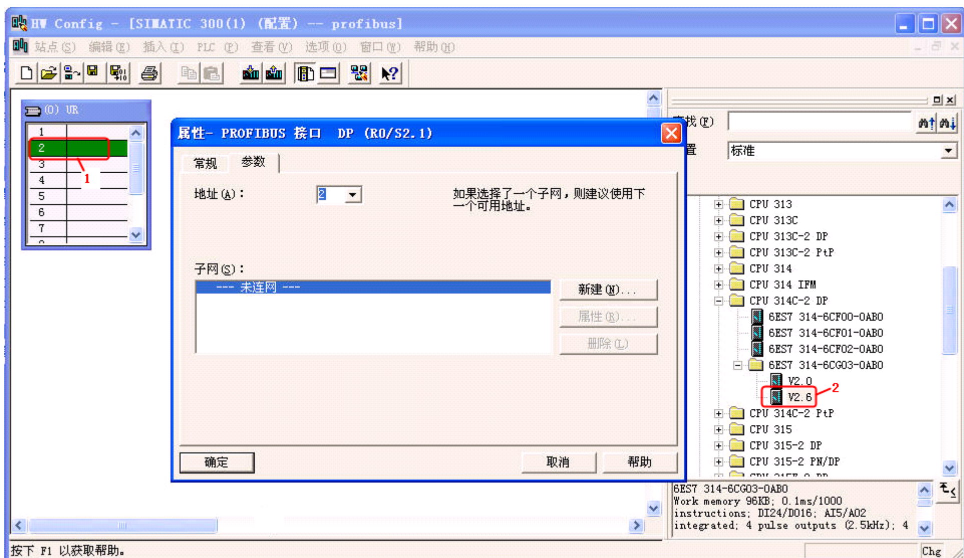


图 6-54　插入 CPU

6）配置网络。双击 2 号槽中的“DP”，弹出“属性-DP”对话框，单击“属性”按钮，再弹出“属性-PROFIBUS 接口”对话框，如图 6-55 所示；单击“新建”按钮，再弹出“属



性-新建子网 PROFIBUS”对话框，如图 6-56 所示；选定传输率为“1.5Mbps”和配置文件为“DP”，单击“确定”按钮，如图 6-57 所示。从站便可以挂在 PROFIBUS 总线上。

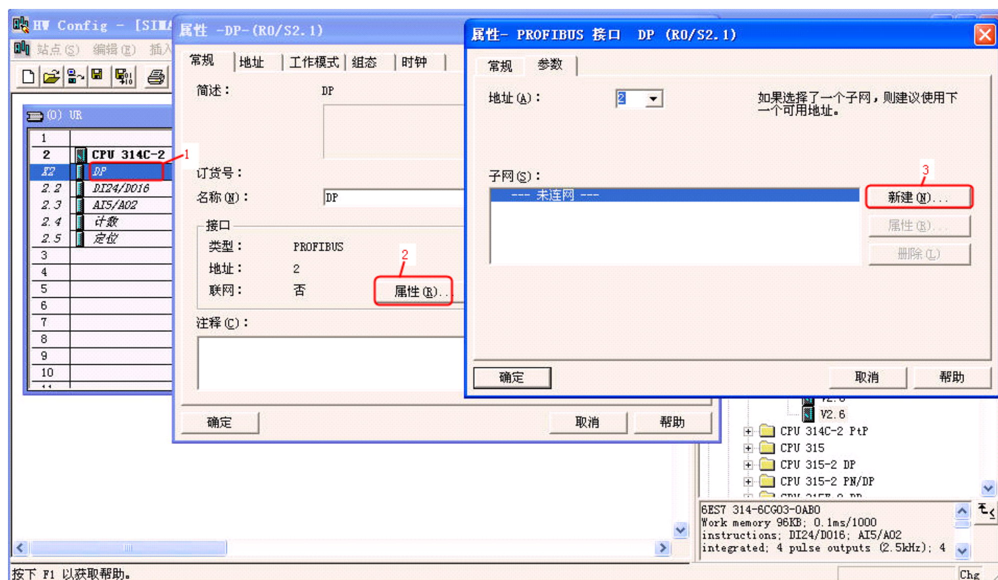


图 6-55 新建网络网络

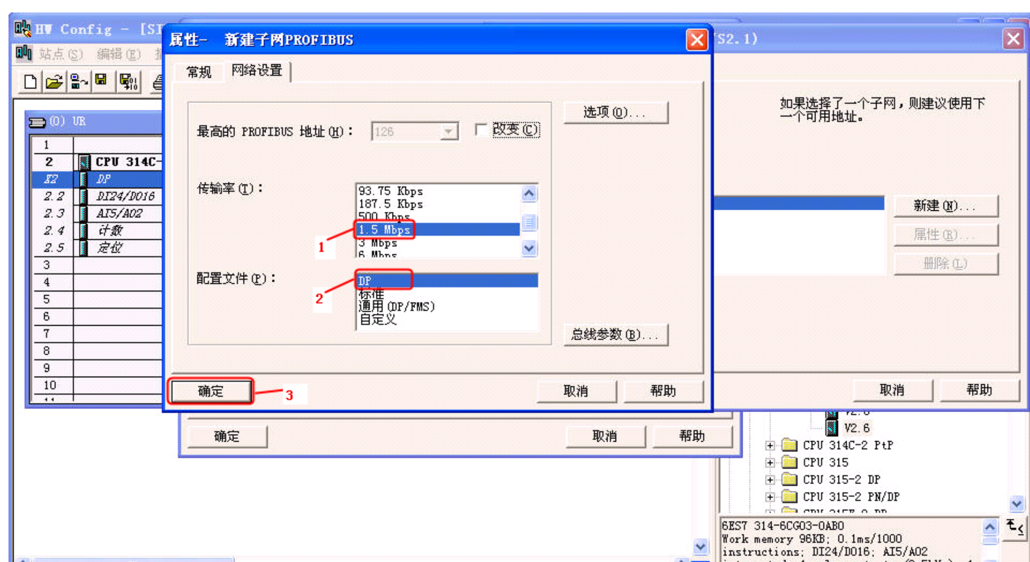


图 6-56 设置通信参数

7）修改 I/O 起始地址。双击 2 号槽中的“DI24/DO16”，弹出“属性-DI24/DO16”对话框，如图 6-58 所示；去掉“系统默认”前的“√”，在“输入”和“输出”的“开始”中输入“0”，单击“确定”按钮，如图 6-59 所示。这个步骤目的主要是为了使程序中输入和输出的起始地址都从“0”开始，这样更加符合我们的习惯，若没有这个步骤，也是可行的，但程序中输入和输出的起始地址都从“124”开始，不方便。





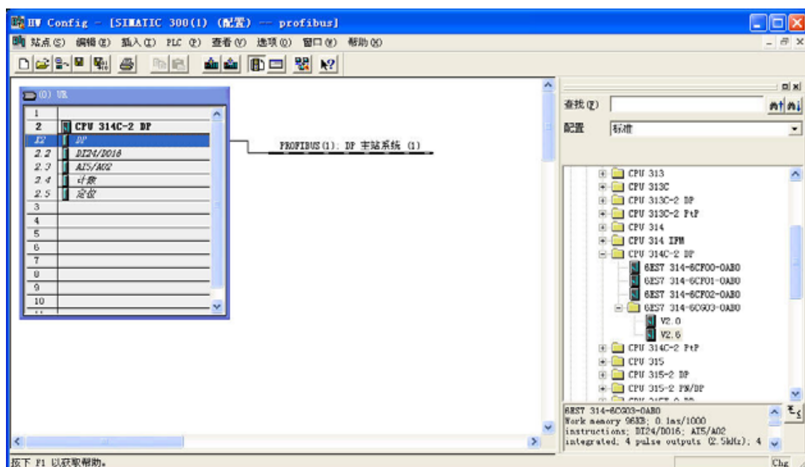


图 6-57　配置网络

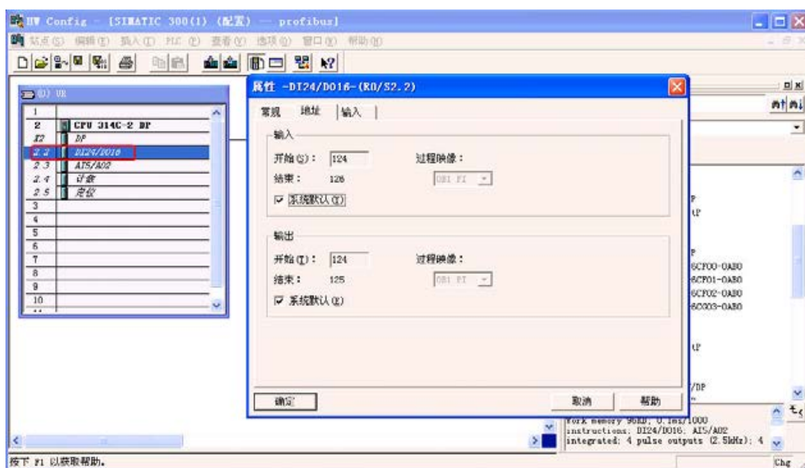


图 6-58　修改 I/O 起始地址（1）

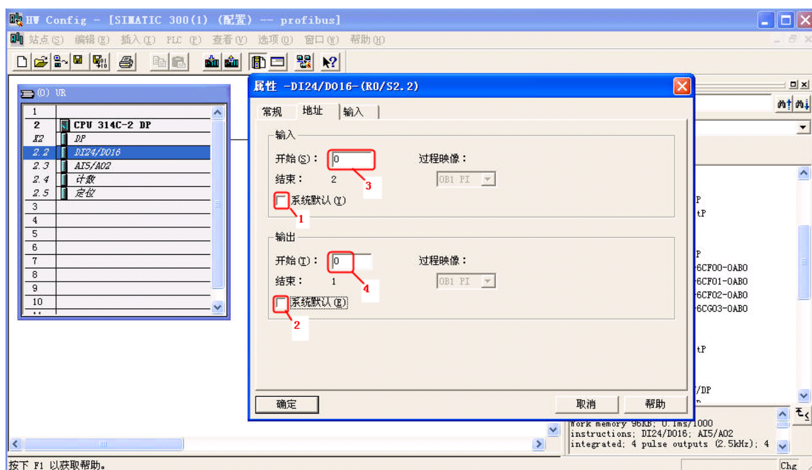


图 6-59　修改 I/O 起始地址（2）



8）配置从站地址。先选中“PROFIBUS”，再展开项目，先后展开“PROFIBUS-DP”→“Additional Field Device”→“PLC”→“SIMATIC”，再双击“EM277 PROFIBUS-DP”，弹出“属性-PROFIBUS 接口”对话框，将地址改为“3”，最后单击“确定”按钮，如图 6-60 所示。

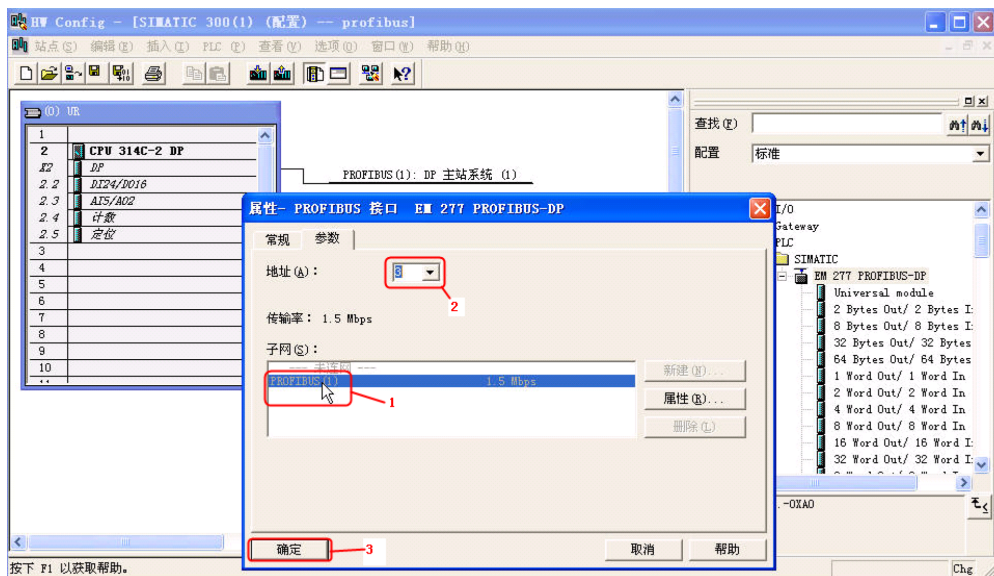


图 6-60 配置从站地址

9）分配从站通信数据存储区。先选中 3 号站，展开项目“EM277 PROFIBUS-DP”，再双击“1 Word In/1 Word Out”，如图 6-61 所示。当然也可以选其他的选项，这个选项的含义是：每次主站接收信息为 1 个字，发送的信息也为 1 个字。

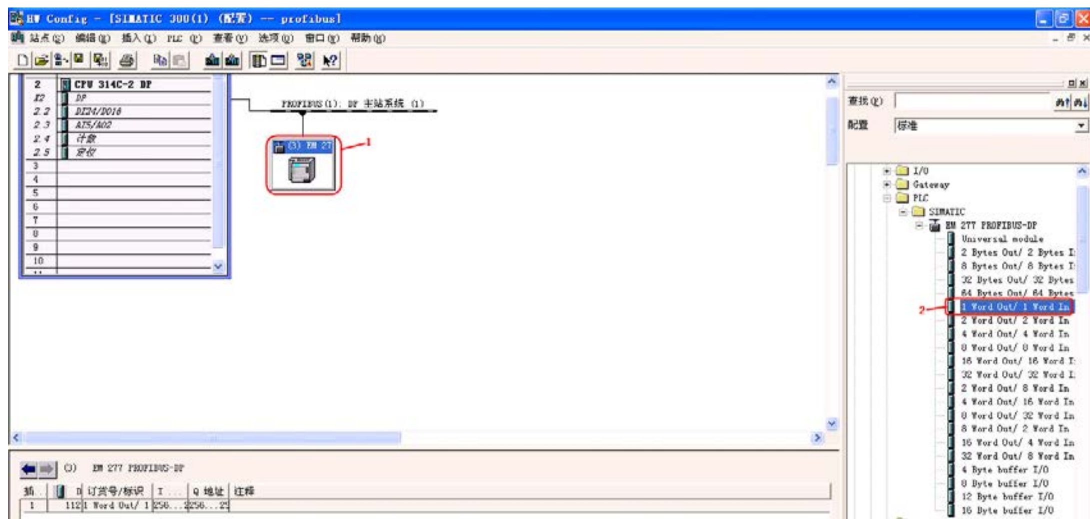


图 6-61 分配从站通信数据存储区

10）修改通信数据发送区和接收数据区起始地址。先选中 3 号站下数据的接收和发送区，双击之，弹出 “属性-DP 从站”，对话框，如图 6−62 所示；再在输入的启动地址中输入“3”，输出启动地址中输入“2”，如图 6−63 所示，再单击“确定”按钮。这样做的目的是为了使后续程序的输入/输出地址更加符合我们的习惯，这个步骤可以没有。

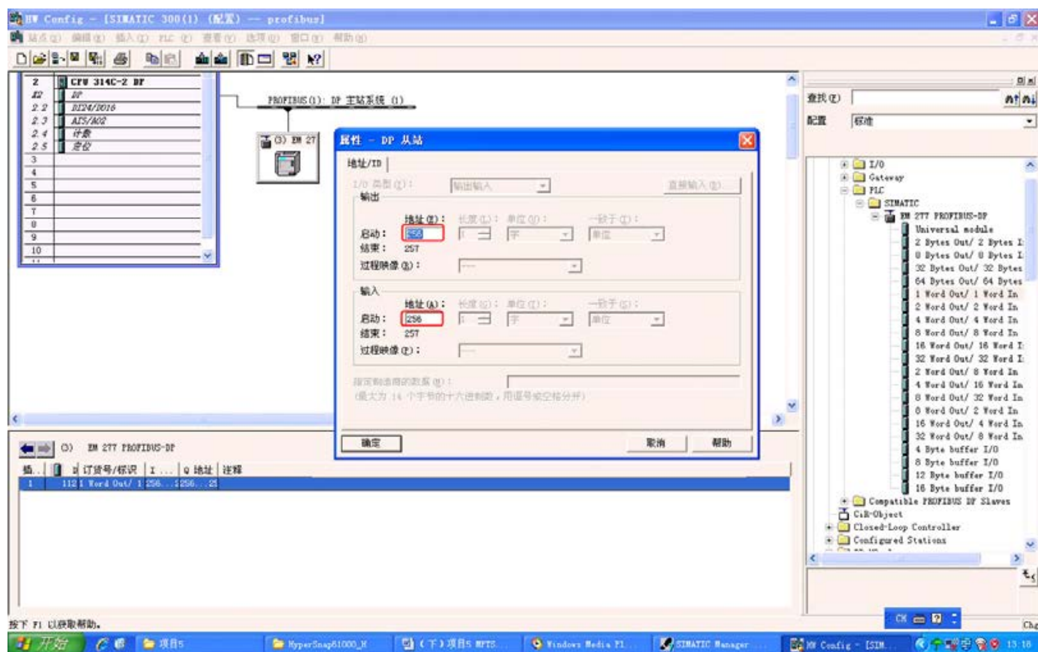


图 6−62　修改通信数据发送区和接收数据区起始地址（1）

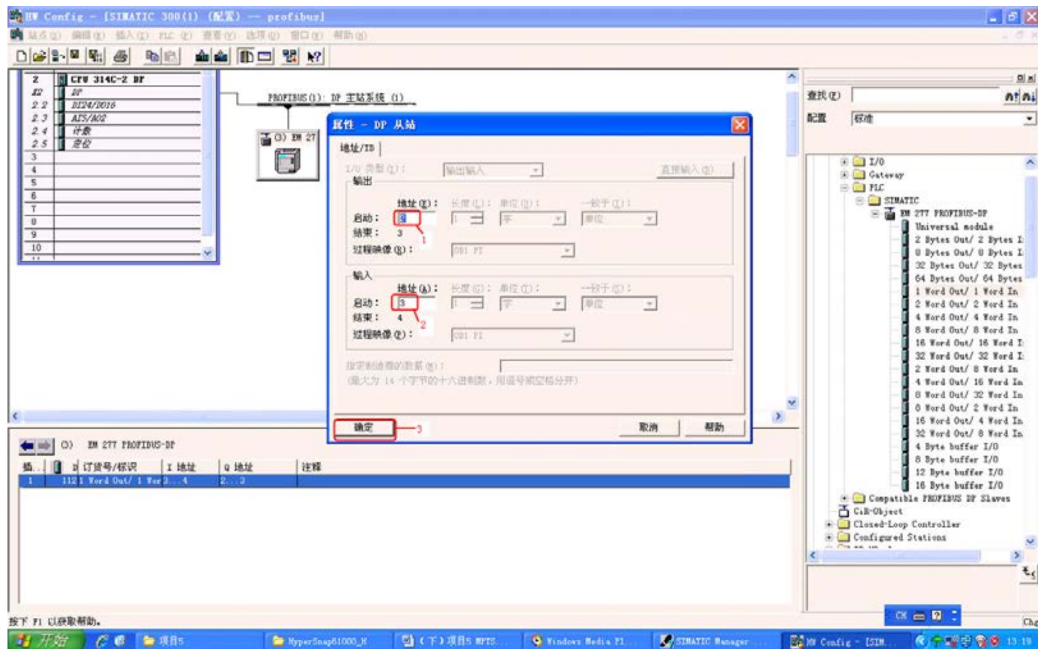


图 6−63　修改通信数据发送区和接收数据区起始地址（2）



11）下载硬件组态。到目前为止，已经完成了硬件的组态，单击“保存和编译”按钮，若有错误，则会显示，没有错误，系统将自动保存硬件组态；接着单击“下载”按钮，系统将硬件配置下载到 PLC 中。下载硬件的步骤是不可缺少的，否则前面所做的硬件配置的工作都是徒劳的，但保存和编译步骤可以省略，因为单击下载按钮也可以起到这个作用。

12）打开块并编译程序。激活“SIMATIC Manager-profibus”界面，展开工程“profibus”，选中“块”，如图 6-64 所示。单击“OB1”，弹出“属性-组织块”对话框，再单击“确定”按钮，如图 6-65 所示。之后弹出“LAD/STL/FBD”界面，实际上是程序编辑界面，在此界面上，输入如图 6-66 所示的程序。

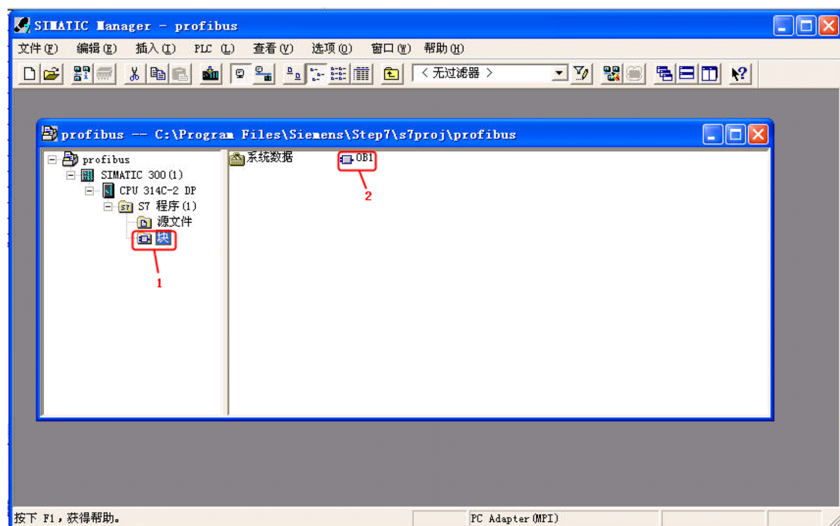


图 6-64 打开 OB1（1）

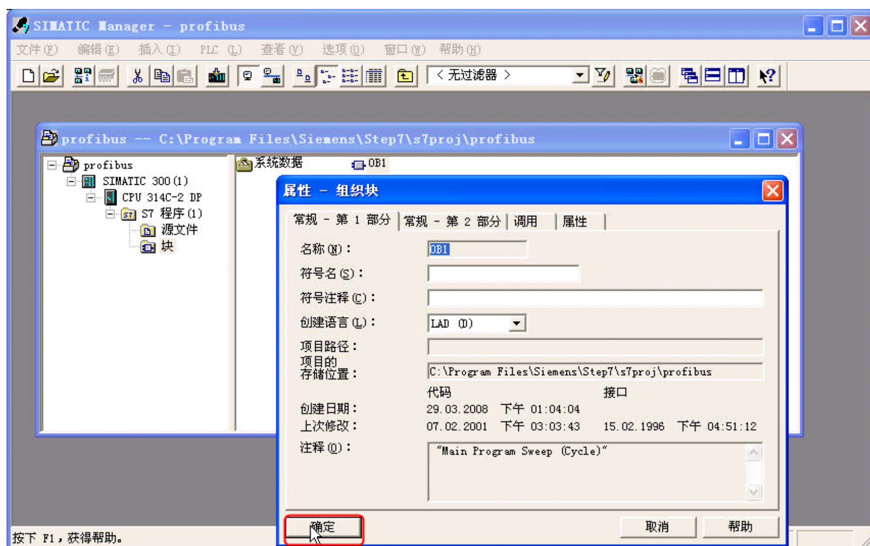


图 6-65 打开 OB1（2）

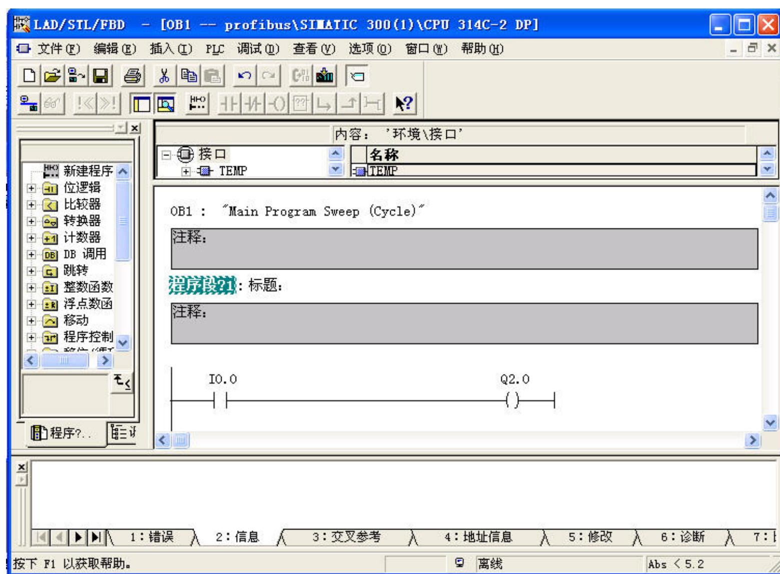


图 6-66　CPU 314C-2DP 的程序

3．编写程序

（1）编写主站的程序

按照以上步骤进行硬件组态后，主站和从站的通信数据发送区和接收数据区就可以进行数据通信了，主站和从站的发送区和接收数据区对应关系见表 6-8。

表 6-8　主站和从站的发送区和接收数据区对应关系

序号	主站 S7-300	对应关系	S7-200 从站
1	QW2	⟶	VW0
2	IW3	⟵	VW2

主站将信息存入 QW2 中，发送到从站的 VW0 数据存储区，那么主站的发送数据区为什么是 QW2 呢？因为 CPU 314C-2DP 自身是 16 点数字输出占用了 QW0，因此不可能是 QW0，QW2 是在前面的序（9）中设定的。当然也可以设定为其他的单元，但不可以设定为 QW0。从站的接收区默认为 VW0，从站的发送区默认为 VW2，这个单元是可以在硬件组态时更改的，请读者参考西门子的相关手册。从站的信息可以通过 VW2 送到主站的 IW3。注意，务必要将组态后的硬件和编译后软件全部下载到 PLC 中。

（2）编写从站程序

在桌面上双击快捷键，打开软件 STEP 7 MicroWin，在梯形图中输入如图 6-67 的程序；再将程序下载到从站 PLC 中。图中的程序含义是：当从站收到信号时，VW0 大于 1，M10.0 自锁输出，Q0.0 以 1s 的频率闪烁。

4．硬件连接

主站 CPU 314C-2DP 有两个 DB9 接口，一个是 MPI 接口，它主要用于下载程序（也可作为 MPI 通信使用），另一个 DB9 接口是 DP 口，PROFIBUS 通信使用这个接口。从站为



CPU 226CN+EM277，EM277 是 PROFIBUS 专用模块，这个模块上面 DB9 接口为 DP 口。主站的 DP 口和从站的 DP 口用专用的 PROFIBUS 电缆和专用网络接头相连，主站和从站的硬件连线如图 6-46 所示。

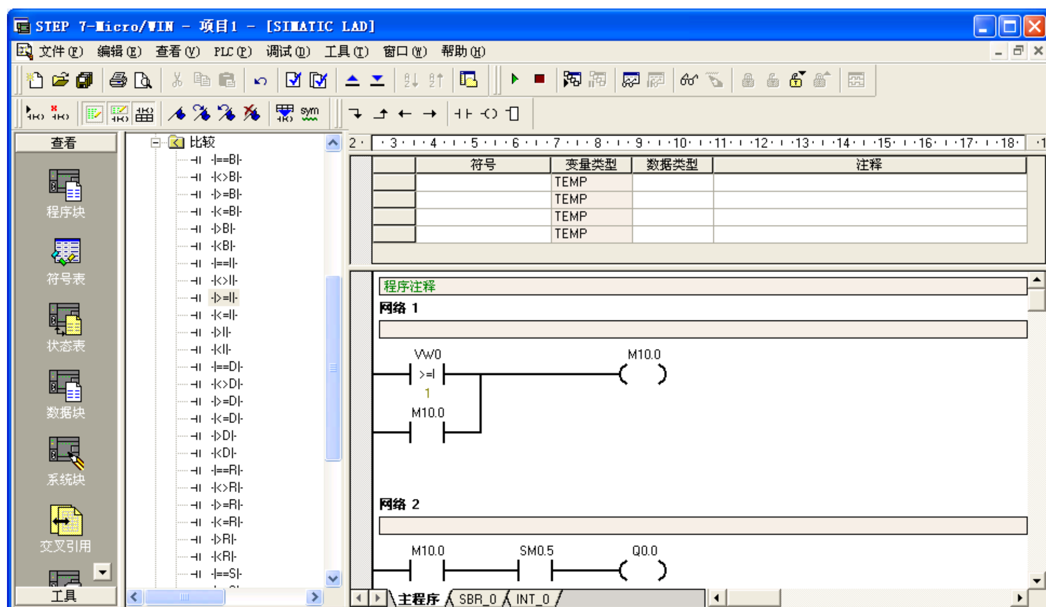


图 6-67　CPU 226CN 的程序

PROFIBUS 电缆是两线屏蔽双绞线，两根线为 A 线和 B 线，电线塑料皮上印刷有 A、B 字母，A 线与网络接头上的 A 端子相连，B 线与网络接头上的 B 端子相连即可。B 线实际与 DB9 的第 3 针相连，A 线实际与 DB9 的第 8 针相连。

【关键点】在前述的硬件组态中已经设定从站为第三站，因此在通信前，必须要将 EM277 的“站号”选择旋钮旋转到“3”的位置，否则，通信不能成功。此外，完成设定 EM277 的站地址后，必须将 EM277 断电，新设定的站地址才能生效。从站网络连接器的终端电阻应置于“on”，如图 6-68 所示。若要置于“off”，只要将拨钮拨向“off”一侧即可。

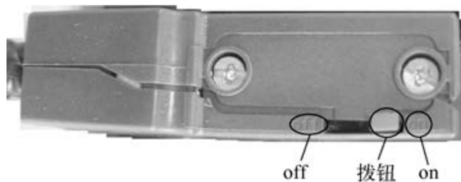


图 6-68　网络连接器的终端电阻置于“on”

5．软硬件调试

用 PROFIBUS 的电缆将 S7-300 PLC 的 DP 口与 EM277 的 DP 口相连，并将 S7-300 PLC 端的网络连接器上的拨钮拨到“off”，并将 EM277 端的网络连接器上的拨钮拨到“on”上。再将程序下载到 PLC 中。最后将两台 PLC 的运行状态从“STOP”都拨到“RUN”上。

6.3.3　S7-300 PLC与S7-300 PLC间的现场总线通信

S7-300 PLC 与 S7-300 PLC 间的现场总线通信同 S7-300 PLC 与 S7-200 PLC 间的现场总线通信有所不同，有的 S7-300 PLC CPU 自带有 DP 通信口（如 CPU 314C-2DP），进行 PROFIBUS 通信时，只需要将两台 S7-300 PLC CPU 的 DP 通信口用 PROFIBUS 通信电缆连接即可。而有的 S7-300 PLC CPU 没有自带的 DP 通信口（如 CPU 314C），要进行 PROFIBUS 通信时，还必须配置 DP 接口模块（CP342-5）。以下仅以两台 CPU 314C-2DP 之间 PROFIBUS 通信为例介绍 S7-300 PLC 与 S7-300 PLC 间的 PROFIBUS 现场总线通信。

【例 6-6】 有两台设备，分别由一台 CPU 314C-2DP 控制，从设备 1 上的 CPU 314C-2DP 发出起/停控制命令，设备 2 的 CPU 314C-2DP 收到命令后，对设备 2 进行起停控制，同时设备 1 上的 CPU 314C-2DP 监控设备 2 的运行状态。

【解】

1．主要软硬件配置

① 1 套 STEP 7 V5.4 SP4。

② 2 台 CPU 314C-2DP。

③ 1 根编程电缆（或者 CP5611 卡）。

④ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置图和 PLC 接线图如图 6-69 和图 6-70 所示。

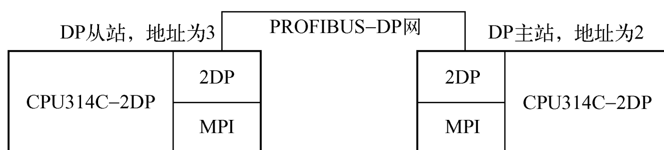


图 6-69　PROFIBUS 现场总线硬件配置图

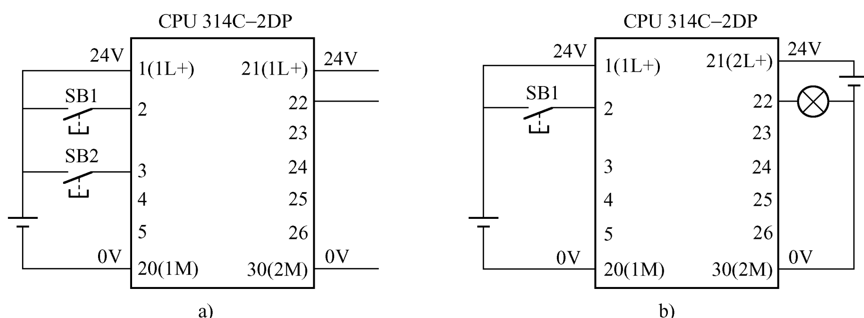


图 6-70　PROFIBUS 现场总线通信 PLC 接线图

a) 主站接线图　b) 从站接线图

2．硬件组态

1）新建工程并插入站点。首先新建一个工程，本例为“Profibus-s7300”，如图 6-71 所示。再在工程中插入两个站点，本例为“Client”和“Server”，共插入 2 个站点，并将站点重命名为“Client”和“Server”，如图 6-72 所示。



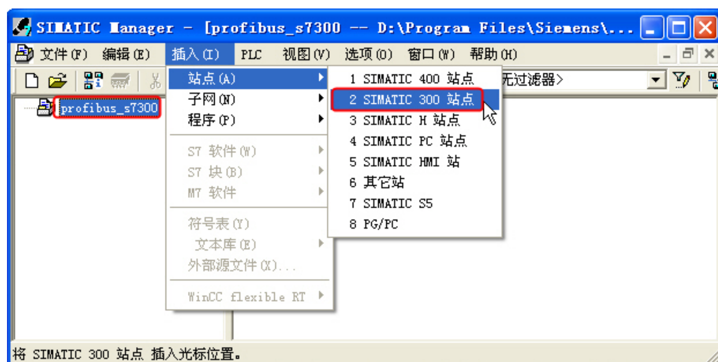


图 6-71　新建工程并插入站点

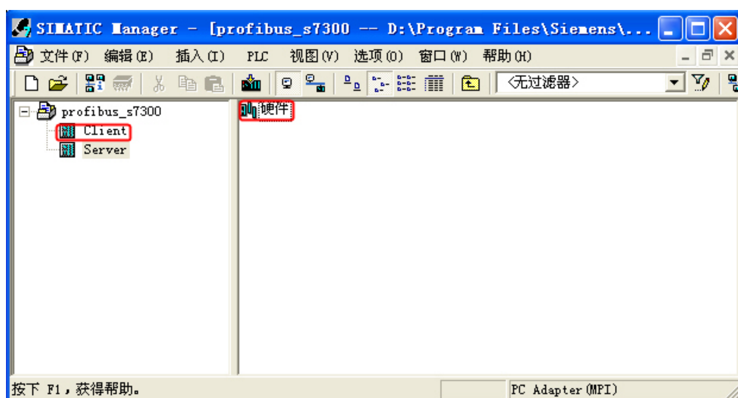


图 6-72　插入站点并重命名

2）插入导轨。如图 6-72 所示，选中从站 “Client”，双击“硬件”，弹出如图 6-73 所示的界面，双击导轨“Rail”，弹出“1”处的导轨。

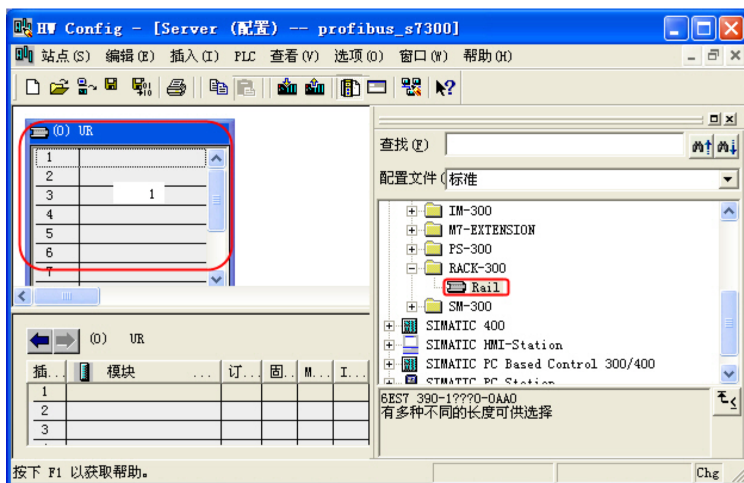


图 6-73　插入导轨

3）插入 CPU 模块。如图 6-74 所示，先选中导轨的 2 号槽位，再展开 CPU 314C-2DP，双击“V2.6”，也可直接用鼠标的左键选中“V2.6”并按住左键不放，直接将 CPU 拖入 2 号槽。

【关键点】CPU 314C-2DP 有 4 个产品型号，读者在组态时，一定要注意 CPU 314C-2DP 机壳上印刷的产品型号要与组态选择的产品型号一致。另外，“314-6CG03-0AB0”还有两个版本，在组态时也要注意与机壳上印刷的一致，否则会出错。

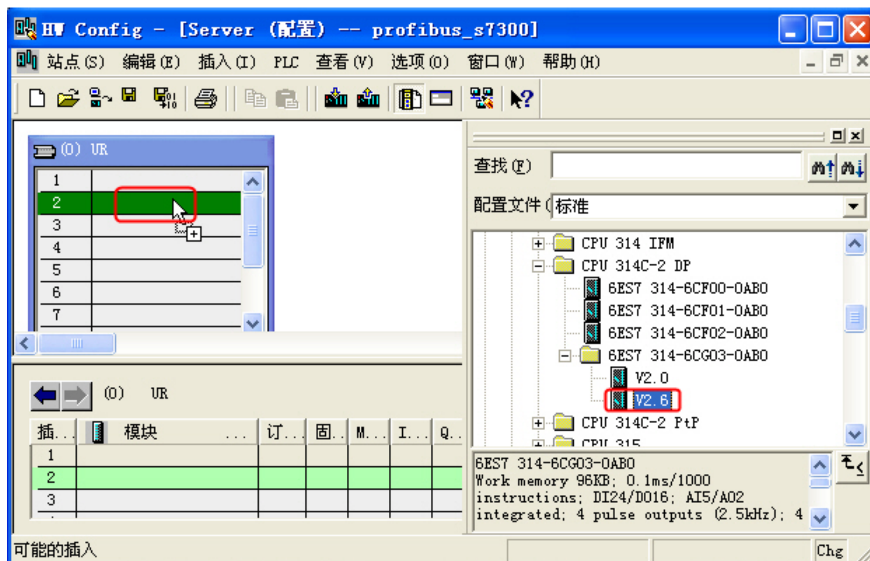


图 6-74　插入 CPU 模块

4）新建 PROFIBUS 网络。如图 6-75 所示，先选定从站的站地址为“3”，再单击“新建”按钮，弹出如图 6-76 所示的界面。

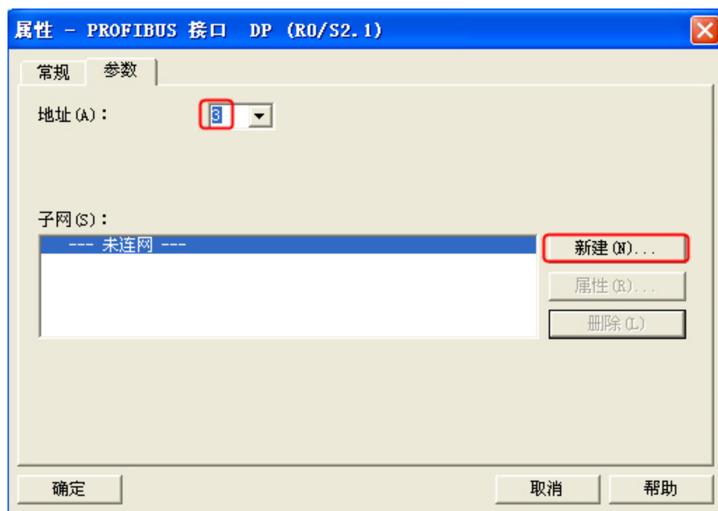


图 6-75　新建 PROFIBUS 网络



5）选择通信的波特率。如图 6-76 所示，先选定 PROFIBUS 的通信的波特率为“1.5Mbps”，再单击“确定”按钮，弹出如图 6-77 所示的界面。

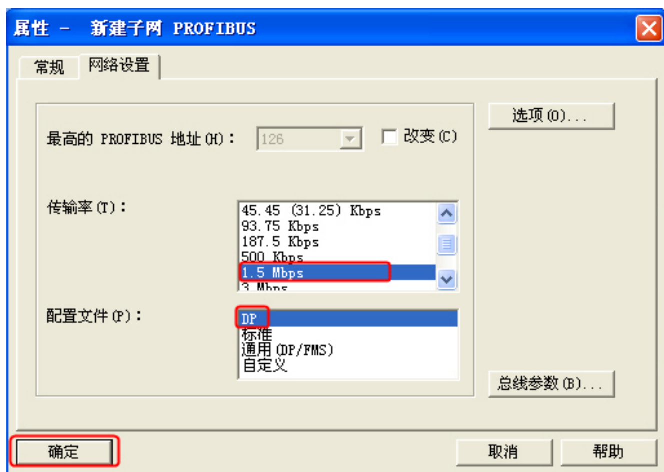


图 6-76 选择通信的波特率

6）选择操作模式。如图 6-77 所示，先双击“1”处的 DP，再选择操作模式为“DP 从站”模式选项，再选定“组态”选项卡，弹出如图 6-78 所示的界面。

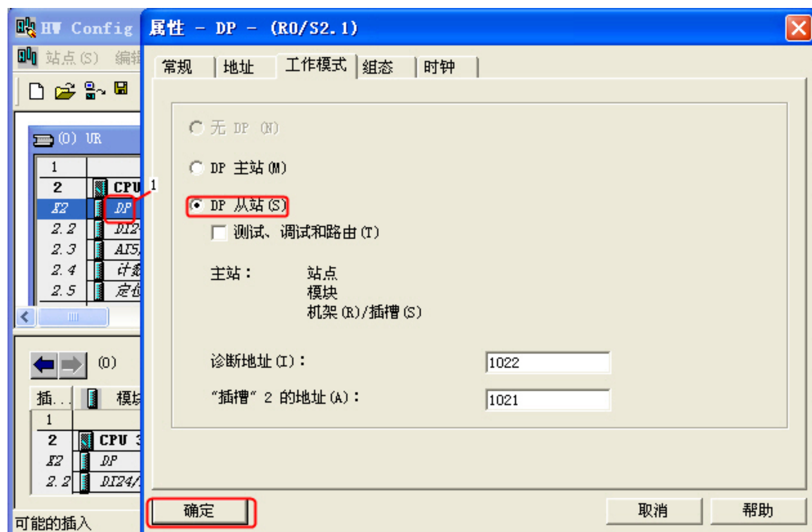


图 6-77 工作模式选择

7）组态接收区和接收区的数据。如图 6-78 所示，先单击“新建”按钮，弹出如图 6-79 所示的界面，定义从站 3 的接收区的地址为“3”（实际就是 QB3），再单击“确定”按钮，接收区数据定义完成。再单击图 6-78 中的“新建”按钮，弹出如图 6-80 所示的界面，定义从站 3 的发送区的地址为“3”（实际就是 IB3），再单击“确定”按钮，发送区数据定义完成。弹出如图 6-81 所示的界面，单击“确定”按钮，从站的发送接收区数据组态完成。

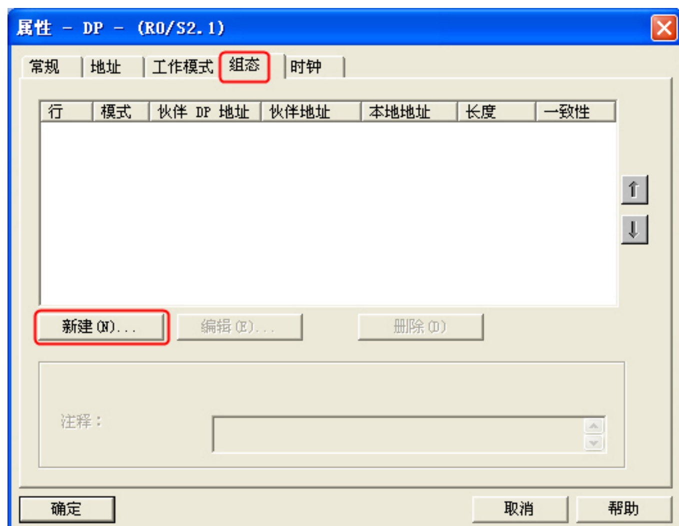


图 6-78　组态通信接口数据区

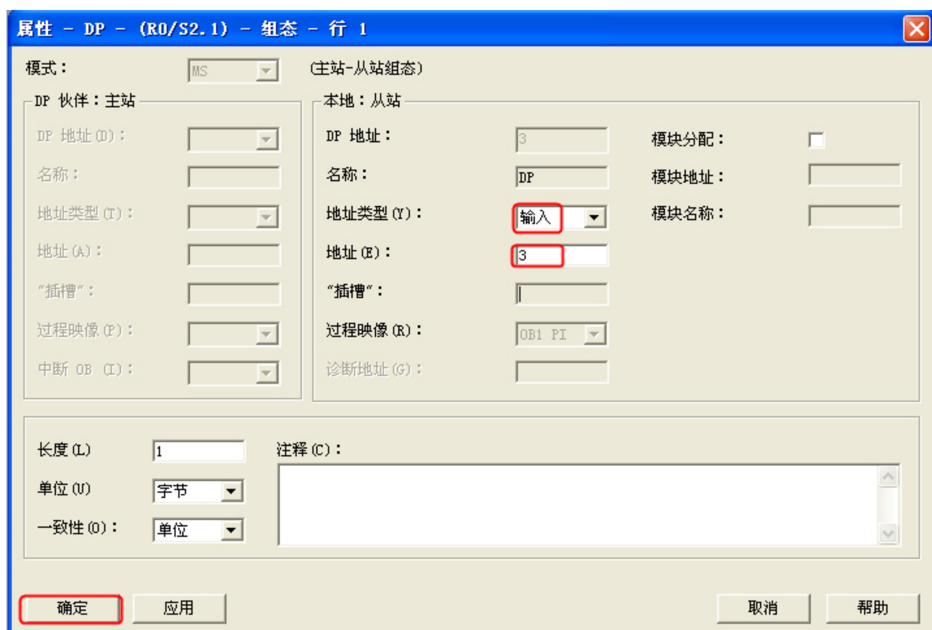


图 6-79　组态接收区数据

8）主站组态时插入导轨和插入 CPU 与从站组态类似，不再重复，以下从选择通信波特率开始讲解，如图 6-82 所示，先选定主站 2 的通信地址为“2”，再选定通信的波特率为“1.5Mbps”，单击”确定”按钮，弹出如图 6-83 所示的界面。

9）将从站 3 挂到 PROFIBUS 网络上。如图 6-83 所示，先用鼠标选中 PROFIBUS 网络的“1”处，再双击“CPU 31x”，弹出如图 6-84 所示的界面。

10）激活从站 3。如图 6-84 所示，单击“连接”按钮，弹出如图 6-85 所示的界面。



图 6-80　组态发送区数据

图 6-81　从站数据区组态完成

图 6-82　选择通信波特率

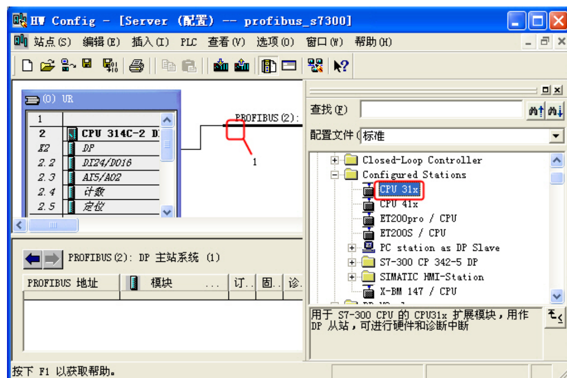


图 6-83　将从站 3 挂到 PROFIBUS 网络上

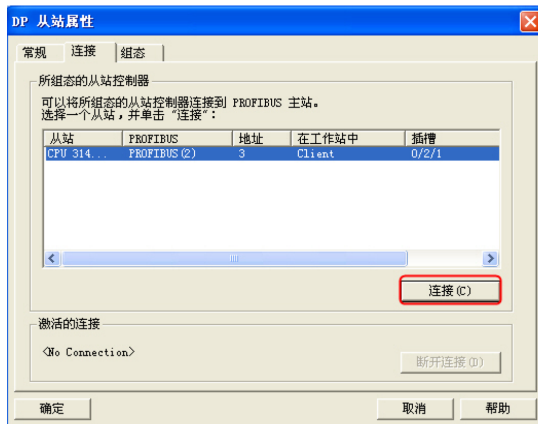


图 6-84　激活从站 3

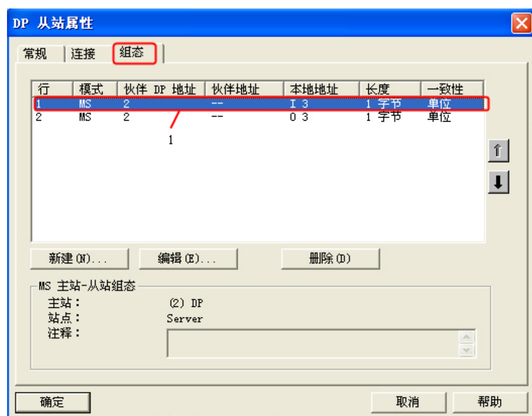


图 6-85　组态主站通信接口数据区

11）组态主站通信接口数据区。如图 6-85 所示，选中“组态”选项卡，再双击“1”处，弹出如图 6-86 所示的界面。先选择地址类型为发送数据，再选定地址为“3”（实际就是 QB3），单击“确定”按钮，发送数据区组态完成。接收数据区的组态方法类似，只需要将如图 6-87 中地址类型选择为接收数据，再选定地址为“3”（实际就是 IB3），单击“确定”按钮。



DP 从站属性 – 组态 – 行 3

模式： MS （主站-从站组态）

DP 伙伴：主站

DP 地址(D)： 2

名称： DP

地址类型(T)： 输出

地址(A)： 3

"插槽"：

过程映像(P)： OB1 PI

中断 OB (I)：

本地：从站

DP 地址： 3

名称： DP

地址类型(Y)： 输入

地址(E)： 3

"插槽"：

过程映像(R)： ---

诊断地址(G)：

模块分配：

模块地址：

模块名称：

长度(L) 1

单位(U) 字节

一致性(O)： 单位

注释(C)：

确定 应用 取消 帮助

图 6-86 组态发送数据区

DP 从站属性 – 组态 – 行 2

模式： MS （主站-从站组态）

DP 伙伴：主站

DP 地址(D)： 2

名称： DP

地址类型(T)： 输入

地址(A)： 3

"插槽"： 5

过程映像(P)： OB1 PI

中断 OB (I)：

本地：从站

DP 地址： 3

名称： DP

地址类型(Y)： 输出

地址(E)： 3

"插槽"： 5

过程映像(R)： OB1 PI

诊断地址(G)：

模块分配：

模块地址：

模块名称：

长度(L) 1

单位(U) 字节

一致性(O)： 单位

注释(C)：

确定 应用 关闭 帮助

图 6-87 组态接收数据区

12）硬件组态完成。在如图 6-88 所示中，单击”确定”按钮，弹出如图 6-89 所示的界面。至此，主站的组态已经完成。

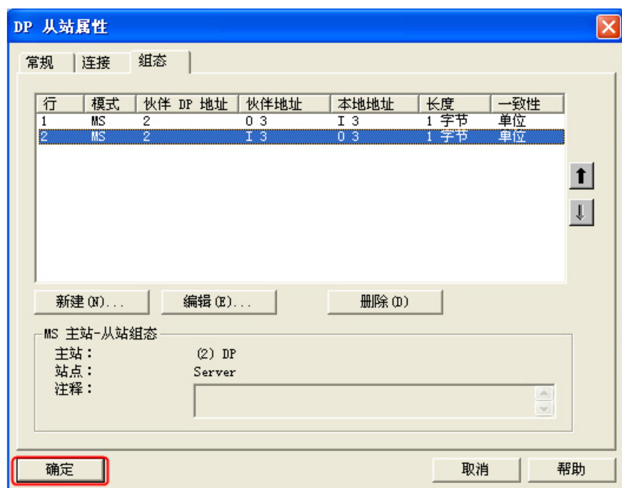


图 6-88　硬件组态完成（1）

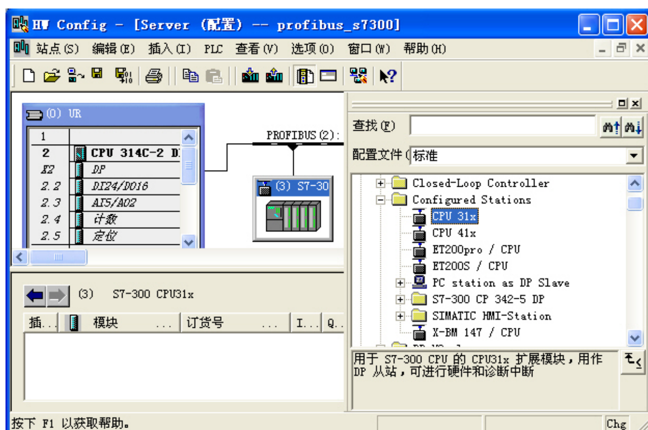


图 6-89　硬件组态完成（2）

13）单击按钮，保存和编译硬件组态。硬件组态完毕。

【关键点】在进行硬件组态时，主站和从站的波特率要相等，主站和从站的地址不能相同，本例的主站地址为 2，从站的地址为 3。最为关键的是：先对从站组态，再对主站进行组态。

3．编写主站程序

S7-300 PLC 与 S7-300 PLC 间的现场总线通信的程序编写有很多种方法，本例是最为简单的一种方法。从图 6-88 中，很容易看出主站 2 和从站 3 的数据交换的对应关系，也可参见表 6-9。

表 6-9　主站和从站的发送接收数据区对应关系

序号	主站 S7-300 PLC	对应关系	从站 S7-300 PLC
1	QB3	——→	IB3
2	IB3	←——	QB3



主站的程序如图 6-90 所示。

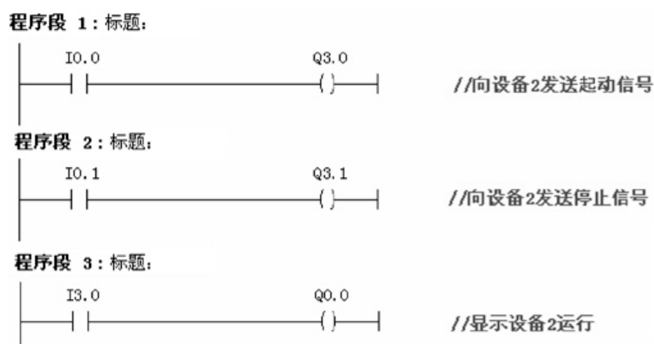


图 6-90　主站程序

4．编写从站程序

从站程序如图 6-91 所示。

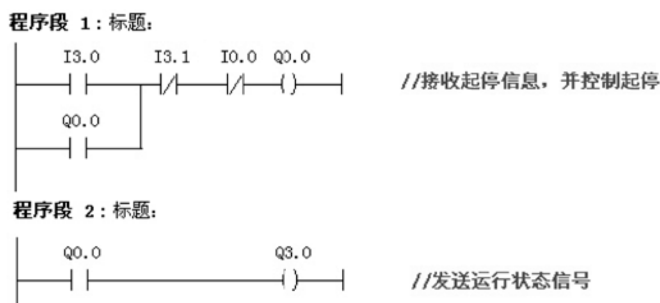


图 6-91　从站程序

6.4　以太网通信及其应用

以太网（Ethernet），指的是由 Xerox 公司创建，并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范。以太网使用 CSMA/CD（载波监听多路访问及冲突检测技术）技术，并以 10Mbit/s 的速率运行在多种类型的电缆上。以太网与 IEEE802·3 系列标准相类似。以太网不是一种具体的网络，而是一种技术规范。

6.4.1　以太网通信基础

1．以太网的历史

以太网的核心思想是：使用公共传输信道。这个思想产生于 1968 年美国的夏威尔大学。

以太网技术的最初进展源自于施乐帕洛阿尔托研究中心的许多先锋技术项目中的一个。人们通常认为以太网发明于 1973 年，以当年罗伯特·梅特卡夫（Robert Metcalfe）给他 PARC

的老板写了一篇有关以太网潜力的备忘录为标志。

1979 年，梅特卡夫成立了 3Com 公司。3Com 联合迪吉多、英特尔和施乐（DEC、Intel 和 Xerox）共同将网络进行标准化、规范化。这个通用的以太网标准于 1980 年 9 月 30 日出台。

2．以太网的分类

以太网分为标准以太网、快速以太网、千兆以太网和万兆以太网。

3．以太网的连接

（1）拓扑结构

1）星形。如图 6-92a 所示，管理方便、容易扩展、需要专用的网络设备作为网络的核心节点、需要更多的网线和对核心设备的可靠性要求高。采用专用的网络设备（如集线器或交换机）作为核心节点，通过双绞线将局域网中的各台主机连接到核心节点上，这就形成了星形结构。星形网络虽然需要的线缆比总线型多，但布线和连接器比总线型的要便宜。此外，星形拓扑可以通过级联的方式很方便地将网络扩展到很大的规模，因此得到了广泛的应用，被绝大部分的以太网所采用。

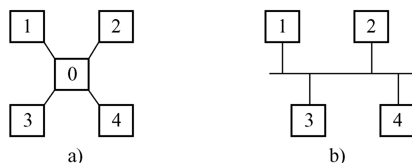


图 6-92　拓扑图
a) 星形结构　b) 总线型结构

2）总线型。如图 6-92b 所示，所需的电缆较少、价格便宜、管理成本高、不易隔离故障点、采用共享的访问机制，易造成网络拥塞。早期以太网多使用总线型的拓扑结构，采用同轴缆作为传输介质，连接简单，通常在小规模的网络中不需要专用的网络设备，但由于它存在的固有缺陷，已经逐渐被以集线器和交换机为核心的星形网络所代替。

此外，还有环形、网状和蜂窝状等拓扑结构。

（2）接口的工作模式

以太网卡可以工作在两种模式下：半双工和全双工。

（3）传输介质

以太网可以采用多种连接介质，包括同轴缆、双绞线和光纤等。其中双绞线多用于从主机到集线器或交换机的连接，而光纤则主要用于交换机间的级联和交换机到路由器间的点到点链路上。同轴缆作为早期的主要连接介质已经逐渐趋于淘汰。

总之，以太网是目前世界上最为流行的拓扑标准之一，具有传播速度高、网络资源丰富、系统功能强大、安装简单和使用维修方便等优点。

4．工业以太网通信简介

（1）初识工业以太网

所谓工业以太网，通俗地讲就是应用于工业的以太网，是指其在技术上与商用以太网（IEEE802.3 标准）兼容，但材质的选用、产品的强度和适用性方面应能满足工业现场的需要。工业以太网技术的优点表现在：以太网技术应用广泛，为所有的编程语言所支持；软硬件资源丰富；易于与 Internet 连接，实现办公自动化网络与工业控制网络的无缝连接；通信速度快；可持续发展的空间大等。

虽然以太网有众多的优点，但作为信息技术基础的 Ethernet 是为 IT 领域应用而开发的，在工业自动化领域只得到有限应用，这是由于：



1）Ethernet 采用 CSMA/CD 碰撞检测方式，在网络负荷较重时，网络的确定性（Determinism）不能满足工业控制的实时要求。

2）Ethernet 所用的接插件、集线器、交换机和电缆等是为办公室应用而设计，不符合工业现场恶劣环境要求。

3）在工程环境中，Ethernet 抗干扰（EMI）性能较差。若用于危险场合，以太网不具备本质安全性能。

4）Ethernet 网还不具备通过信号线向现场仪表供电的性能。

随着信息网络技术的发展，上述问题正在迅速得到解决。为促进 Ethernet 在工业领域的应用，国际上成立了工业以太网协会（Industrial Ethernet Association，IEA）。

（2）网络电缆接法

用于 Ethernet 的双绞线有 8 芯和 4 芯两种，双绞线的电缆连线方式也有两种，即正线（标准 568B）和反线（标准 568A），其中正线也称为直通线，反线也称为交叉线。正线接线如图 6-93 所示，两端线序一样，从下至上线序是：白橙，橙，白绿，蓝，白蓝，绿，白棕，棕。也就是 568B 标准。反线接线如图 6-94 所示，一端为正线的线序，另一端为反线的线序。也就是 568A 标准。对于千兆以太网，用 8 芯双绞线，但接法不同于以上所述的接法，请参考有关文献。

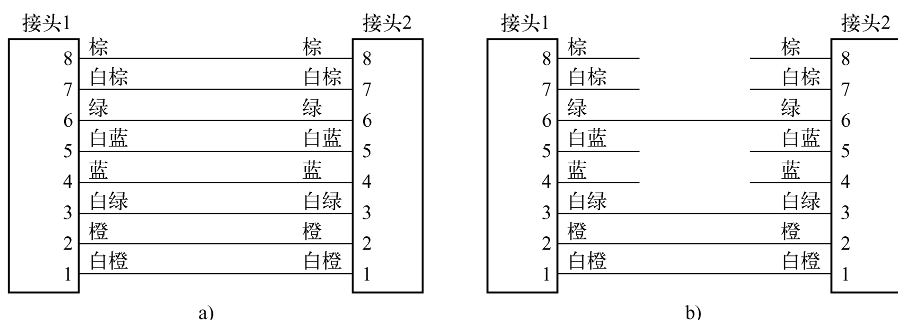


图 6-93　双绞线正线接线图

a) 8 芯线　b) 4 芯线

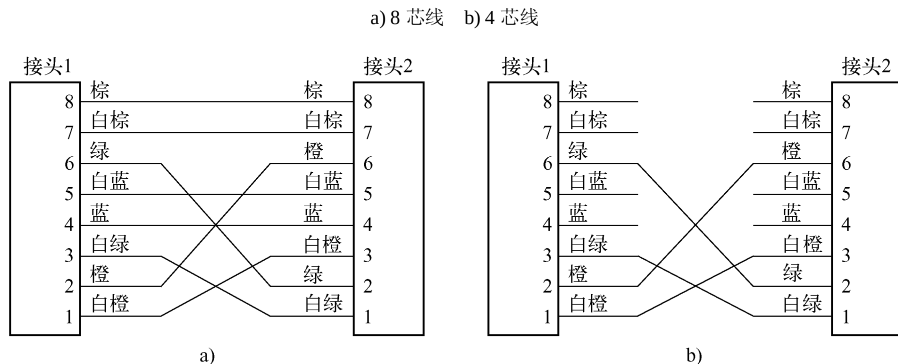


图 6-94　双绞线反线接线图

a) 8 芯线　b) 4 芯线

对于 4 芯的双绞线，只用连接头上的（常称为水晶接头）1、2、3 和 6 四个引脚。西门子的 PROFINET 工业以太网采用 4 芯的双绞线。

常见的采用正线连接的有：计算机（PC）与集线器（HUB）、计算机（PC）与交换机（SWITCH）、PLC 与交换机（SWITCH）、PLC 与集线器（HUB）。

常见的采用反线连接的有：计算机（PC）与计算机（PC）、PLC 与 PLC。

6.4.2　S7-200 PLC与S7-300 PLC间的以太网通信

S7-200 PLC 与 S7-300 PLC 间的以太网通信，S7-300 PLC 可以作为服务器端或者客户端，S7-200 PLC 也可以作为服务器端或者客户端，但 S7-300 PLC 配置有以太网模块（如 CP343-1 Lean），S7-300 PLC 只能作为服务器端。以 S7-300 PLC 作为服务器端，S7-200 PLC 作为客户端为例，介绍 S7-200 PLC 与 S7-300 PLC 间的以太网通信。

【例 6-7】 当 S7-300 PLC 服务器端上发出一个起停信号时，客户端 S7-200 PLC 收到信号，并对一台电动机进行起停控制。

1．本例软硬件配置

S7-200 PLC 与 S7-300 PLC 间的以太网通信硬件配置如图 6-95 所示。

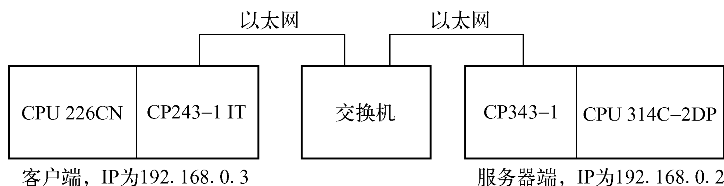


图 6-95　S7-200 PLC 与 S7-300 PLC 间的以太网通信硬件配置图

① 1 台 CPU 226 CN。
② 1 台 CP243-1 IT 以太网模块。
③ 1 台 CPU 314C-2DP。
④ 1 台 CP343-1 Lean 以太网模块。
⑤ 1 台 8 口交换机。
⑥ 2 根带水晶接头的 8 芯双绞线（正线或者反线）。
⑦ 1 套 STEP 7-MICRO/WIN SP6 和 1 套 STEP 7 5.4 SP4。
⑧ 1 根 PC/PPI 电缆（USB 口）。
⑨ 个人电脑（含网卡）。

【关键点】 CP343-1 Lean 只能作为服务器端，不能作为客户端，但 CP343-1 IT 模块既可以作为服务器端，又可以作为客户端。

2．配置客户端

（1）打开“以太网向导”

单击“工具”下的子菜单“以太网向导”，弹出“以太网向导”如图 6-96 和 6-97 所示，单击“下一步”按钮。

（2）指定模块位置

在模块位置中输入位置号，本例为“0”，再单击“读取模块”按钮，若读取成功，则模块的信息显示在如图 6-98 的序号“3”处。





图 6-96　打开以太网向导

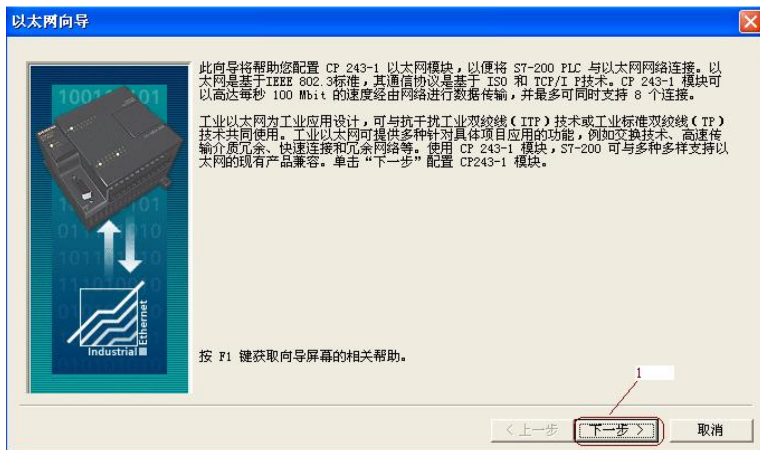


图 6-97　以太网向导初始界面



图 6-98　指定模块位置

（3）指定模块地址

在 IP 地址中输入“192.168.0.3”，在“子网掩码”中输入“255.255.255.0”，网关可以空置，在“为此模块指定通信连接类型”中选择“自动检测通信”选项，最后单击“下一步”按钮，如图 6-99 所示。IP 地址的末位可以为小于等于 255，但除 2 外的所有整数。

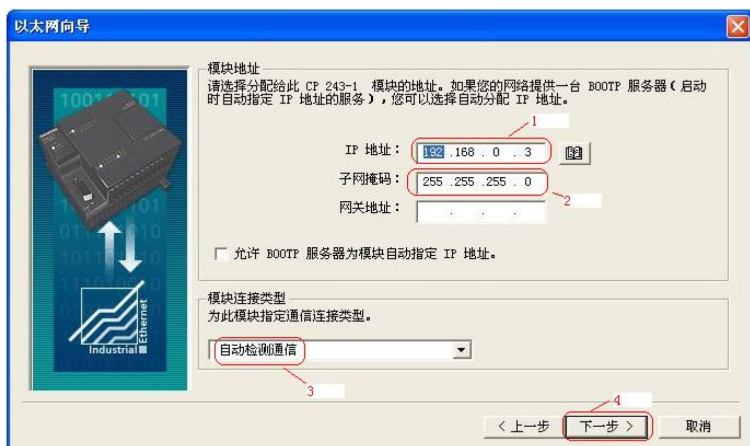


图 6-99　指定模块地址

（4）指定命令字节和连接数目

因为只有两个模块，所以“为此模块配置的连接数”选定为“1”，再单击“下一步”按钮，如图 6-100 所示。

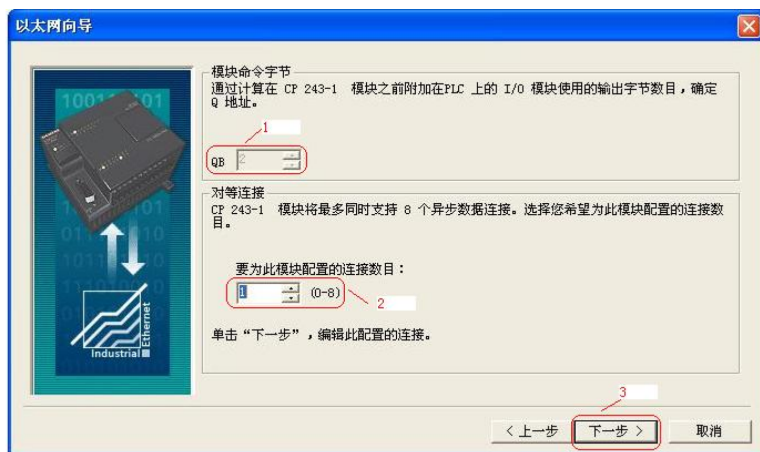


图 6-100　指定命令字节和连接数目

（5）配置连接

将“远程对象”设置为“03.02”（若为 S7-200 PLC，则设定为 10.00），再将“为此连接服务器端的 IP 地址”设定为“192.168.0.2”。单击“确定”按钮，如图 6-101 所示。

单击“新传输”按钮，再单击“确定”按钮，如图 6-102 所示。



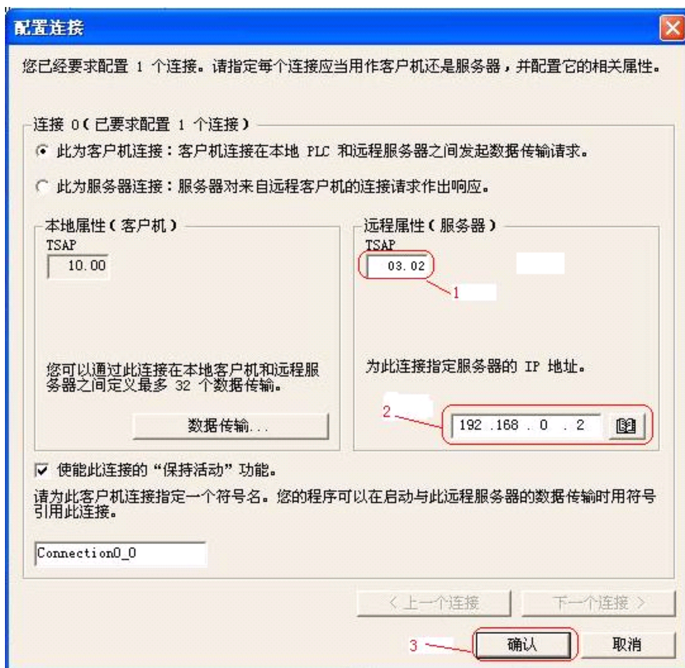


图 6-101　配置连接（1）

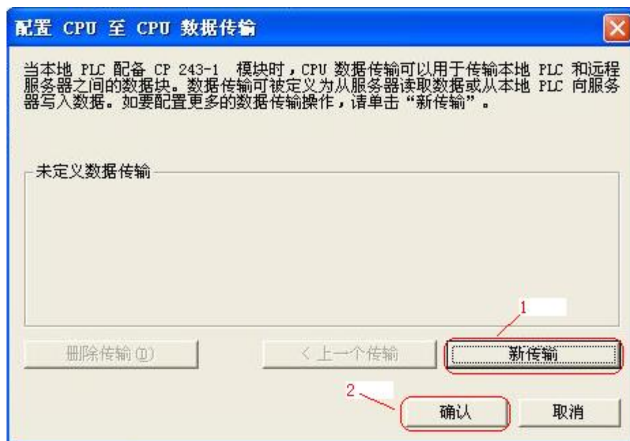


图 6-102　配置连接（2）

如图 6-103 所示，在序“1”处选定“从远程服务器端读取数据”；序“2”处选定为“1”，因为一个字节可以包含 8 个开关量信息，而本例只有一个开关量；序“3”和序“4”处的含义是：将服务器端的 MB0 中的数据传送到客户端的 VB0 中去；再单击“确定”按钮。

（6）CRC 保护与保持现用间隔

如图 6-104 所示，先选定“是，为数据块中的此配置生成 CRC 保护”，再单击“下一步”按钮。

（7）分配配置内存

如图 6-105 所示，先单击“建议地址”，再单击“下一步”按钮。

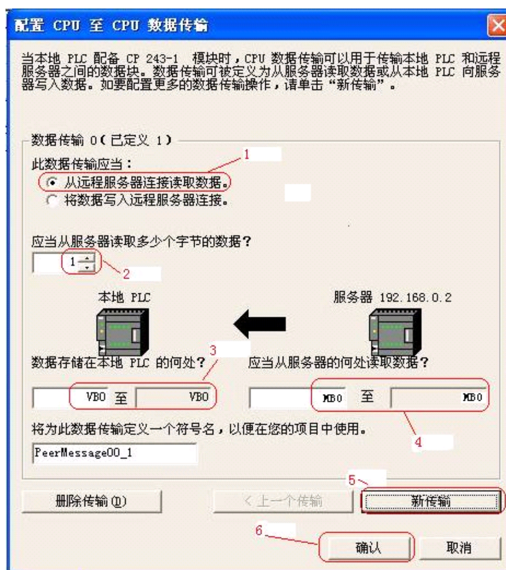


图 6-103　配置连接（3）

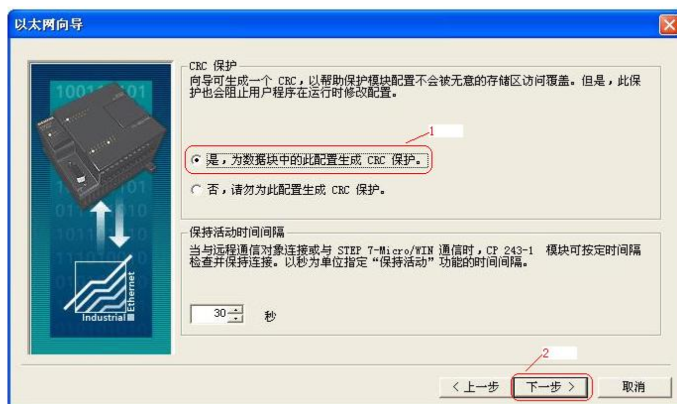


图 6-104　CRC 保护与保持现用间隔



图 6-105　分配配置内存



（8）生成项目部件

如图 6-106 所示，单击“完成”按钮，完成客户端的配置。

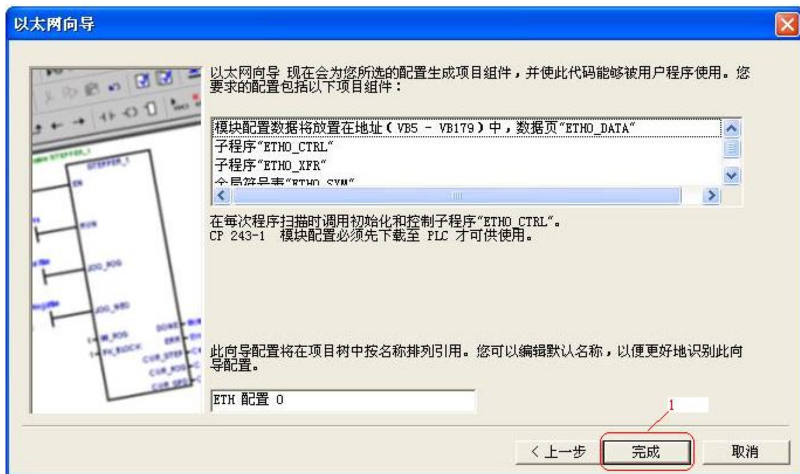


图 6-106　生成项目部件

3．配置服务器端

（1）新建工程

新建工程如图 6-107 所示，选中导轨槽位 4，再双击“CP343-1Lean”模块的“V2.0”版本，弹出 IP 地址设置，如图 6-108 所示。

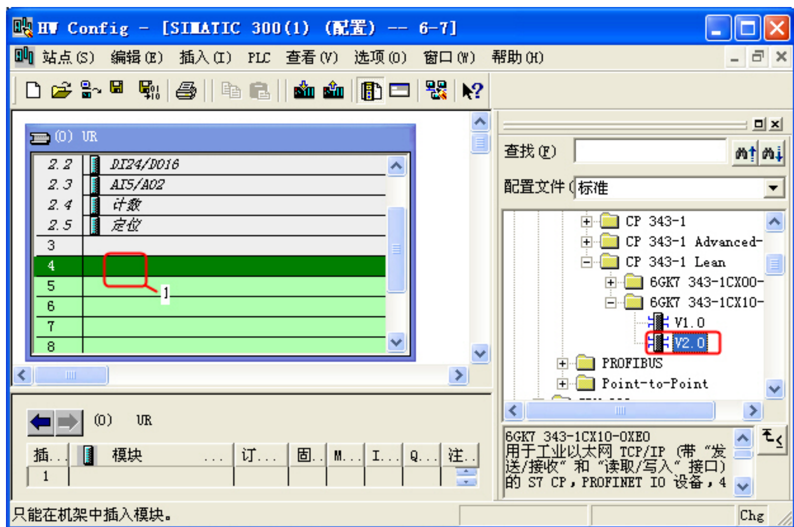


图 6-107　新建工程

（2）IP 地址设置

如图 6-108 所示，在“IP 地址”中填写服务器端地址“192.168.0.2”，在“子网掩码”中填写服务器端地址“255.255.255.0”，再单击“新建”按钮，弹出组件以太网界面，如图 6-109 所示。

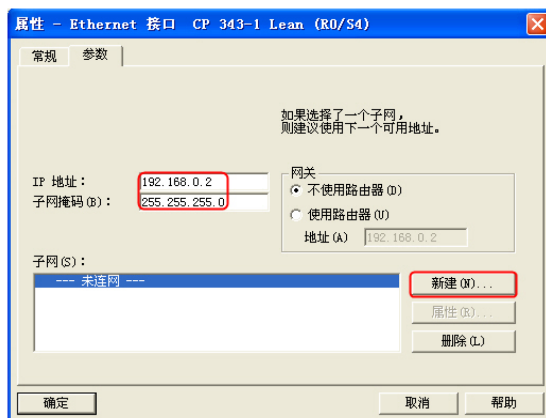


图 6-108　IP 地址设置

（3）组建以太网

如图 6-109 所示，单击“确定”按钮，弹出如图 6-110 所示界面，选中“Ethernet（1）”，单击“确定”按钮，以太网组建完成。

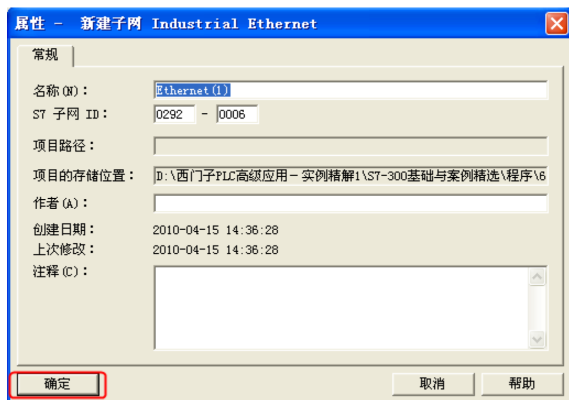


图 6-109　组建以太网（1）

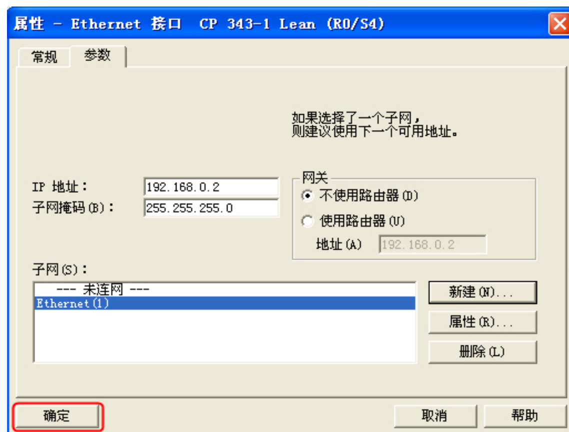


图 6-110　组建以太网（2）



4．编写程序

客户端程序如图 6-111 所示，服务器端程序如图 6-112 所示。

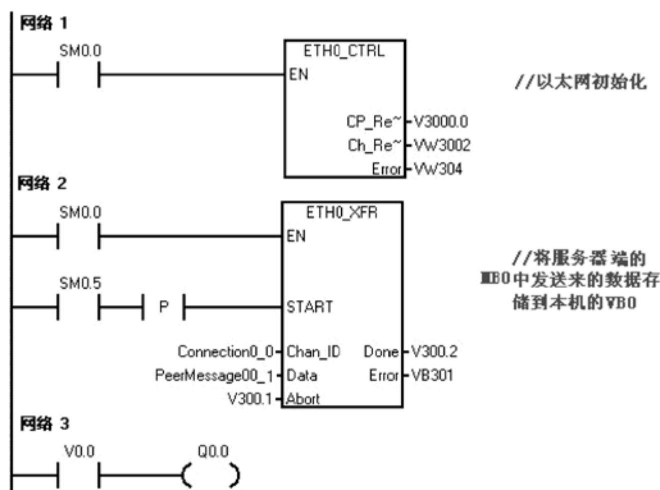


图 6-111　客户端程序

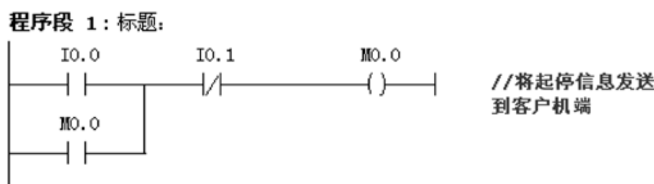


图 6-112　服务器端程序

6.4.3　S7-300 PLC间的以太网通信

1．西门子工业以太网通信方式简介

工业以太网的通信主要利用第 2 层（ISO）和第 4 层（TCP）的协议。以下是西门子以太网的几种通信方式。

（1）ISOTransport（ISO 传输协议）

ISO 传输协议支持基于 ISO 的发送和接收，使得设备（例如 SIMATIC S5 或 PC）在工业以太网上的通信非常容易，该服务支持大数据量的数据传输（最大 8KB）。ISO 数据接收由通信方确认，通过功能块可以看到确认信息。用于 SIMATIC S5 和 SIMATIC S7 的工业以太网连接。

（2）ISO-on-TCP

ISO-on-TCP 支持第 4 层 TCP/IP 协议的开放数据通信。用于支持 SIMATIC S7 和 PC 以及非西门子支持的 TCP/IP 以太网系统。ISO-on-TCP 符合 TCP/IP，但相对于标准的 TCP/IP，还附加了 RFC 1006 协议，RFC 1006 是一个标准协议，该协议描述了如何将 ISO 映射到 TCP 上去。

（3）UDP

UDP（User Datagram Protocol，用户数据报协议），属于第 4 层协议，提供了 S5 兼容通信协议，适用于简单的交叉网络数据传输，没有数据确认报文，不检测数据传输的正确性。UDP 支持基于 UDP 的发送和接收，使得设备（例如 PC 或非西门子公司设备）在工业以太网上的通信非常容易。该协议支持较大数据量的数据传输（最大 2KB），数据可以通过工业以太网或 TCP/IP 网络（拨号网络或因特网）传输。通过 UDP，SIMATIC S7 通过建立 UDP 连接，提供了发送/接收通信功能，与 TCP 不同，UDP 实际上并没有在通信双方建立一个固定的连接。

（4）TCP/IP

TCP/IP 中传输控制协议，支持第 4 层 TCP/IP 协议的开放数据通信。提供了数据流通信，但并不将数据封装成消息块，因而用户并不接收到每一个任务的确认信号。TCP 支持面向 TCP/IP 的 Socket。

TCP 支持给予 TCP/IP 的发送和接收，使得设备（例如 PC 或非西门子设备）在工业以太网上的通信非常容易。该协议支持大数据量的数据传输（最大 8KB），数据可以通过工业以太网或 TCP/IP 网络（拨号网络或因特网）传输。通过 TCP，SIMATIC S7 可以通过建立 TCP 连接来发送/接收数据。

2．S7 通信

S7 通信（S7 Communication）集成在每一个 SIMATIC S7/M7 和 C7 的系统中，属于 OSI 参考模型第 7 层应用层的协议，它独立于各个网络，可以应用于多种网络（MPI、PROFIBUS、工业以太网）。S7 通信通过不断地重复接收数据来保证网络报文的正确。在 SIMATIC S7 中，通过组态建立 S7 连接来实现 S7 通信。在 PC 上，S7 通信需要通过 SAPI-S7 接口函数或 OPC（过程控制用对象链接与嵌入）来实现。

3．实例

以下用两台 S7-300 PLC 的以太网通信为例，介绍 S7-300 PLC 间的以太网通信。

【例 6-8】 当一台 S7-300 PLC（PLC1）上发出一个起停信号时，另一台 S7-300 PLC（PLC2）收到信号，并对一台电动机进行起停控制，PLC2 向 PLC1 反馈电动机的运行状态。

【解】

S7-300 PLC 之间的组态可以采用很多连接方式，如 TCP/IP、ISO-on-TCP 和 S7 Communication 等，以下仅介绍 TCP/IP 连接方式。

（1）软硬件配置

S7-300 PLC 间的以太网通信硬件配置如图 6-113 所示，本例用到的软硬件如下：

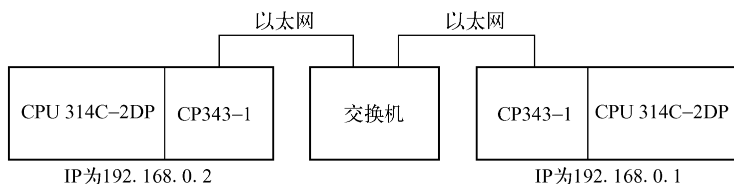


图 6-113　S7-300 PLC 间的以太网通信硬件配置图



① 2 台 CPU 314C-2DP。

② 2 台 CP343-1 以太网模块。

③ 1 根 PC/MPI 适配器（USB 口）。

④ 1 台个人电脑（含网卡）。

⑤ 1 台 8 口交换机。

⑥ 2 根带水晶接头的 8 芯双绞线（正线）。

⑦ 1 套 STEP 7 V5.4 SP4 编程软件。

（2）硬件组态

1）新建工程。插入两个站分别是 PLC1 和 PLC2，每个站点上，配置一台 CP343-1 以太网通信模块，如图 6-114 所示。

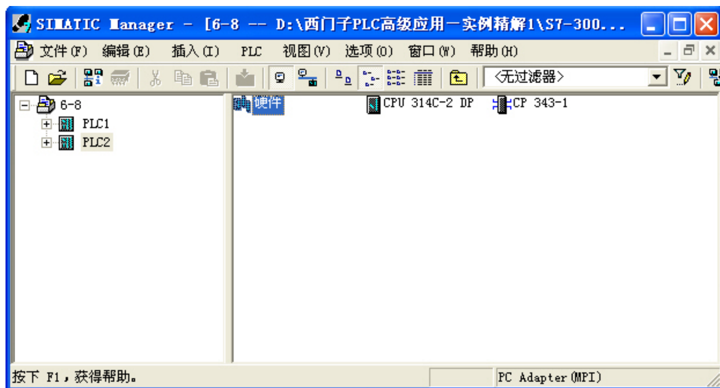


图 6-114　新建工程

2）组态以太网。双击“硬件”，弹出如图 6-115 所示界面，选中“CP 343-1”，并双击之，弹出如图 6-116 所示界面，单击“属性”按钮，弹出如图 6-117 所示界面。

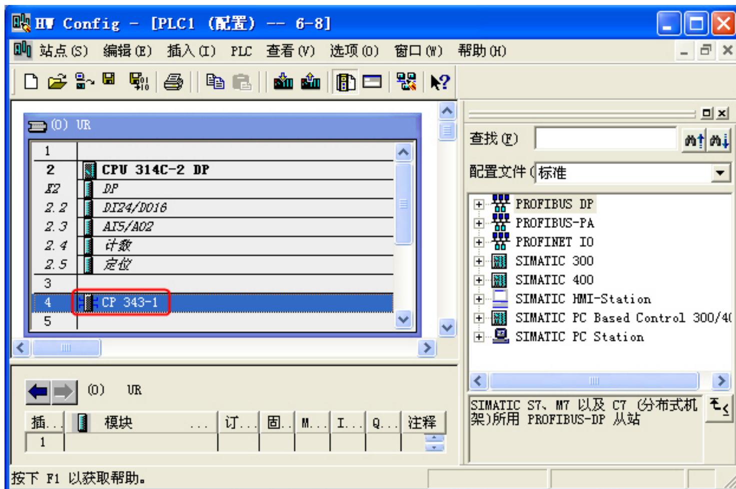


图 6-115　组态以太网（1）

3）新建网络。单击“新建”按钮，弹出如图 6-117 所示界面，单击“确定”按钮，弹出如图 6-118 所示界面。

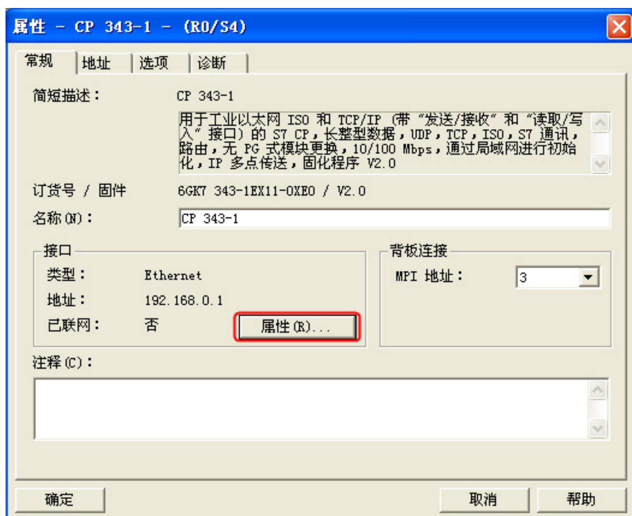


图 6-116　组态以太网（2）

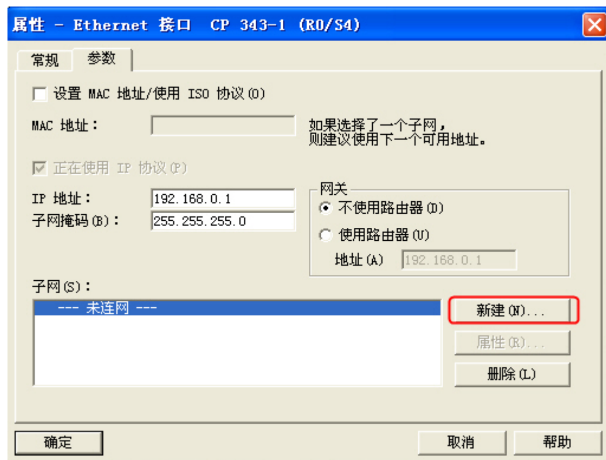


图 6-117　新建以太网（1）

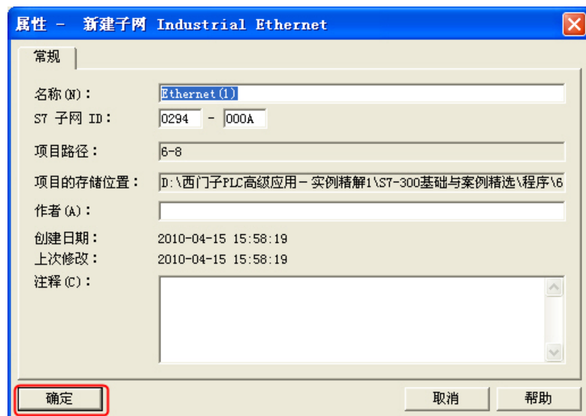


图 6-118　新建以太网（2）



4）设置网络参数。如图 6-119 所示，先选中“Ethernet（1）”，再在“IP 地址”中设置“192.168.0.2”，在“子网掩码”中设置“255.255.255.0”，单击“确定”按钮。

5）采用同样的方法，配置第二个以太网模块的参数，不同之处在于，将“IP 地址”中设置成“192.168.0.1”。

【关键点】同一个网络中，IP 地址是唯一的，绝对不允许重复。

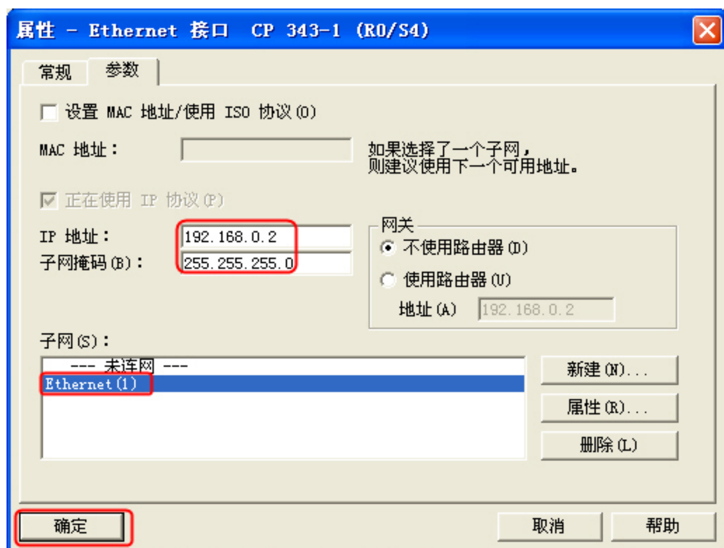


图 6-119　设置网络参数

6）打开网络连接。返回管理界面，如图 6-120 所示，先选中“Ethernet（1）”，再双击“Ethernet（1）”，弹出如图 6-121 所示界面。

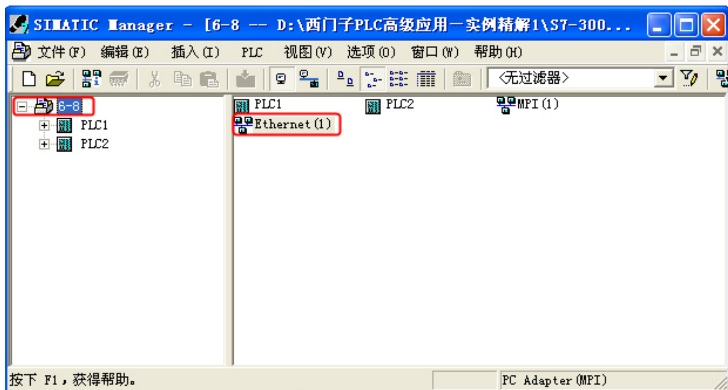


图 6-120　打开网络连接界面

7）组态以太网连接。如图 6-121 所示，先选中“1”处，单击鼠标右键，弹出快捷菜单，再单击“插入新连接”，弹出如图 6-122 所示界面。

【关键点】若一个 PLC 中选择了“Insert New Connection”选项，另一 PLC 则不必激活此项，必须有一台 PLC 选择此选项，以便在通信初始化中起到主动连接的作用。

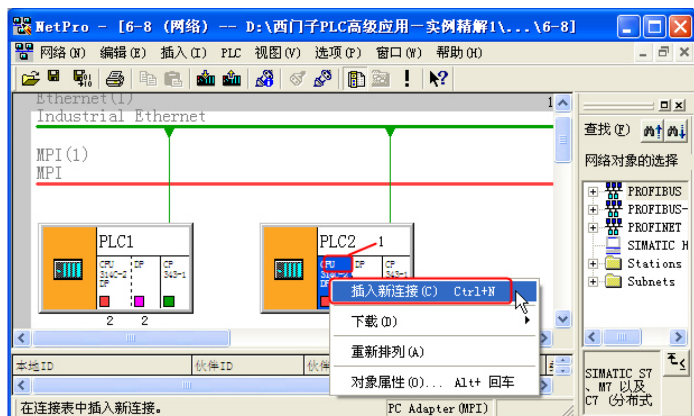


图 6-121　组态以太网连接

8）添加一个 TCP 连接。如图 6-122 所示，先选中“CPU 314C-2DP”，再选择“TCP 连接”，再单击“应用”按钮，弹出如图 6-123 所示界面。

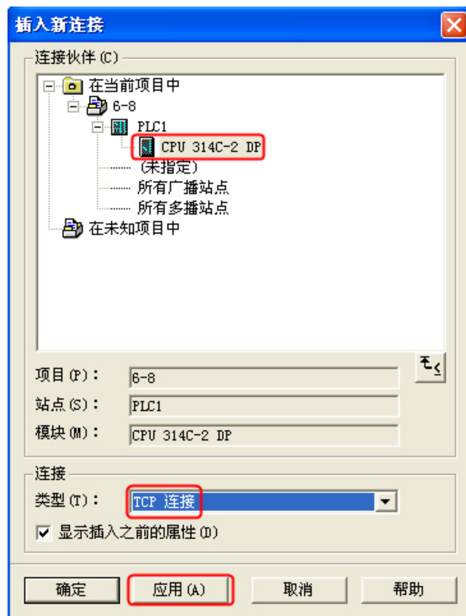


图 6-122　添加一个 TCP 连接

9）设置网络连接参数。如图 6-123 所示，先选择“激活连接”，再单击“确定”按钮。

在如图 6-124 中的“地址”选项卡中可以看到通信双方的 IP 地址，占用的端口号可以自己设置，也可以使用默认值，如 2001。编译后存盘，至此硬件组态完成。

【关键点】图 6-123 中的 ID 是组态时的连接号，LADDR 是模块硬件组态地址，地址相同才能通信，在编程时要用到。

（3）相关指令简介

AG_SEND 块将数据传送给以太网 CP，用于在一个已组态的 ISO 传输连接上进行传输。所选择的数据区可以是一个位存储器区或一个数据块区。当可以在以太网上发送整个用



户数据区时，指示无错执行该功能。AG_SEND 的各项参数见表 6-10。

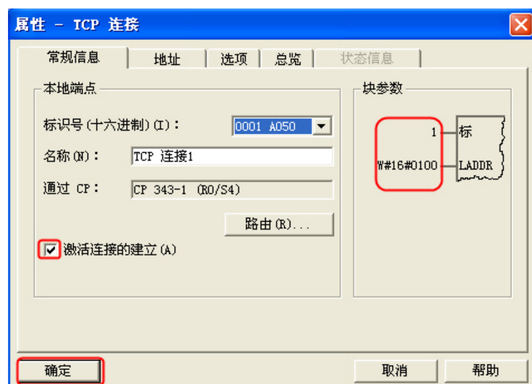


图 6-123　设置网络连接参数

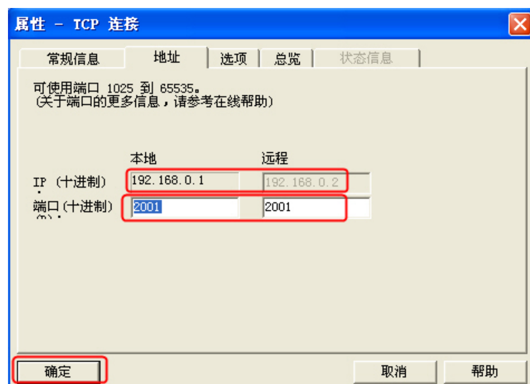


图 6-124　设置 TCP/IP 端口

表 6-10　AG_SEND（FC5）指令格式

LAD	输入 / 输出	含义	数据类型
"AG_SEND" EN　ENO ACT　DONE ID　ERROR LADDR　STATUS SEND LEN	EN	使能	BOOL
	ACT	发送请求	BOOL
	ID	组态时的连接号	INT
	LADDR	模块硬件组态地址	WORD
	SEND	发送的数据区	ANY
	LEN	发送数据长度	INT
	ERROR	错误代码	BOOL
	STATUS	返回数值（如错误值）	WORD
	DONE	发送是否完成	BOOL

AG_RECV 功能（FC）接收从以太网 CP 在已组态的连接上传送的数据。为数据接收指定的数据区可以是一个位存储区或一个数据块区。当可以从以太网 CP 上接收数据时，指示无错执行该功能。AG_RECV 的各项参数见表 6-11。

表 6-11　AG_RECV 指令格式

LAD	输入 / 输出	含义	数据类型
"AG_RECV" EN　ENO ID　NDR LADDR　ERROR RECV　STATUS LEN	EN	使能	BOOL
	ID	组态时的连接号	INT
	LADDR	模块硬件组态地址	WORD
	RECV	接收数据区	ANY
	NDR	接收数据确认	BOOL
	ERROR	错误代码	BOOL
	STATUS	返回数值（如错误值）	WORD
	LEN	接收数据长度	INT

（4）编写程序

在编写程序时，双方都需要编写发送 AG_SEND（FC5）指令和接收 AG_RECV（FC6）指令，PLC1（IP 地址为 192.168.0.2）的程序如图 6-125 所示，PLC2 的程序如图 6-126（IP 地址为 192.168.0.1）所示。

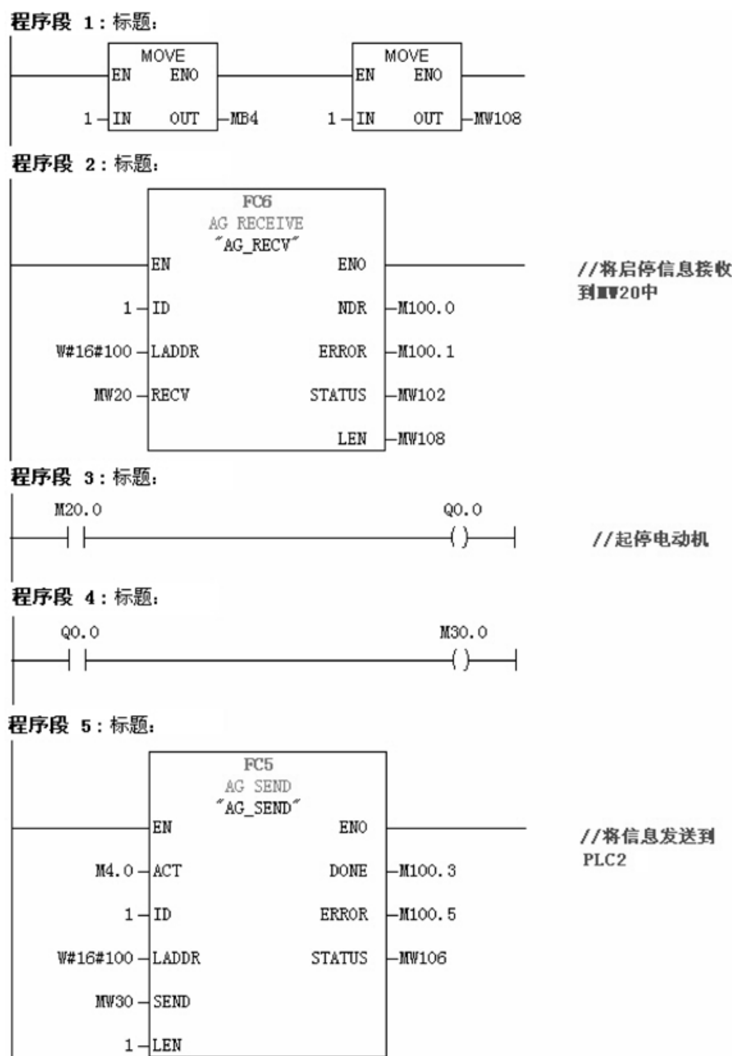


图 6-125　PLC1 中的程序

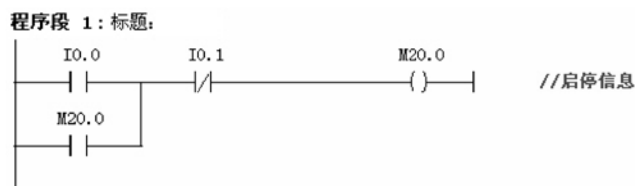
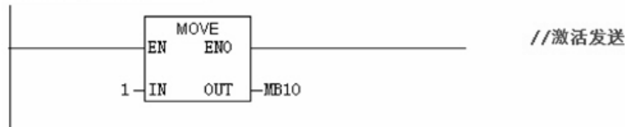
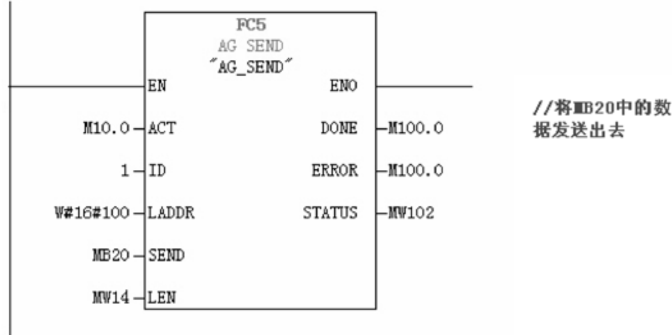


图 6-126　PLC2 中的程序







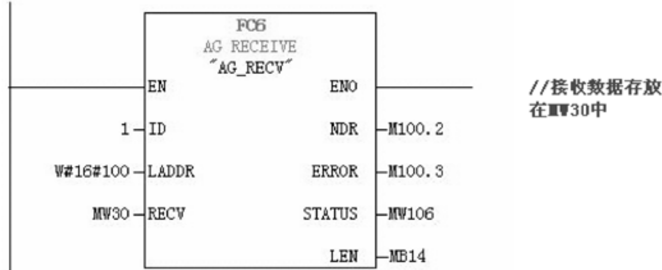




图 6-126　PLC2 中的程序（续）

小结

重点难点总结

通信是 PLC 中的难点，一般初学者不容易掌握，往往通信不成功时，还不知道错误出在何处？其实西门子的每种通信都有固定的模式，只要掌握正确步骤，就可以完成通信。

1．PROFIBUS 现场总线应用很广泛，应重点掌握。

2．工业以太网在稍大的系统中都有应用，特别是车间级及其以上的通信系统中应用较多，应该重点掌握。

3．PROFIBUS 现场总线和工业以太网通信有两个难点：一是通信的组态，步骤较多，初学时容易出错；二是通信专用函数的参数较多，不易理解。

因此要学好本章内容必须下工夫。

习题

1．OSI 模型分为哪几层？各层的作用是什么？

2．PROFIBUS 现场总线的种类有哪些？这些种类的 PROFIBUS 现场总线使用了 OSI 模型的哪几层？

3．西门子 PLC 的常见通信方式有哪几种？

4．有 3 台 CPU 314C-2DP，一台为主站，其余两台为从站，在主站上发出一个起停信号，对从站上控制的电动机进行起停，从站将电动机的起停状态反馈到主站，请编写程序。

5．某设备上有一台 CPU 314C-2DP 为主站，有两台 CPU 226CN 为从站，在主站上发出一个起停信号，对从站上控制的电动机进行起停，从站将电动机的起停状态反馈到主站，请组态硬件并编写程序。

6．何谓串行通信和并行通信？

7．何谓双工、单工和半双工？请举例说明。

第 7 章

S7-300 PLC在电炉温度控制系统中的应用

本章介绍 PID 控制的基本原理以及 PID 控制在电炉温度控制中的应用。

7.1　PID控制简介

在过程控制中，按偏差的比例（P）、积分（I）和微分（D）进行控制的 PID 控制器（也称 PID 调节器）是应用最广泛的一种自动控制器。它具有原理简单，易于实现，适用面广，控制参数相互独立，参数选定比较简单，调整方便等优点；而且在理论上可以证明，对于过程控制的典型对象——“一阶滞后+纯滞后”与“二阶滞后+纯滞后”的控制对象，PID 控制器是一种最优控制。PID 调节规律是连续系统动态品质校正的一种有效方法，它的参数整定方式简便，结构改变灵活（如可为 PI 调节、PD 调节等）。长期以来，PID 控制器被广大科技人员及现场操作人员所采用，并积累了大量的经验。

PID 控制器就是根据系统的误差，利用比例、积分、微分计算出控制量来进行控制。当被控对象的结构和参数不能完全掌握或得不到精确的数学模型时，控制理论的其他技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。即当我们不完全了解一个系统和被控对象或不能通过有效的测量手段来获得系统参数时，最适合采用 PID 控制技术。

1．比例（P）控制

比例控制是一种最简单、最常用的控制方式，如放大器、减速器和弹簧等。比例控制器能立即成比例地响应输入的变化量。但仅有比例控制时，系统输出存在稳态误差（Steady-state error）。

2．积分（I）控制

在积分控制中，控制器的输出量是输入量对时间积累。对一个自动控制系统，如果在进入稳态后存在稳态误差，则称这个控制系统是有稳态误差的或简称有差系统（System with Steady-state Error）。为了消除稳态误差，在控制器中必须引入“积分项”。积分项对误差的运算取决于时间的积分，随着时间的增加，积分项会增大。所以即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大，使稳态误差进一步减小，直到等于零。因此，采用比例+积分（PI）控制器，可以使系统在进入稳态后无稳态误差。

3．微分（D）控制

在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。自动控制系统在克服误差的调节过程中可能会出现振荡甚至失稳。其原因是存在有较大的惯性组件（环节）或有滞后（delay）组件，具有抑制误差的作用，其变化总是落后于误差的变化。解决的办法是使抑制误差的作用的变化“超前”，即在误差接近零时，抑制误差的作用就应该是零。这就是说，在控制器中仅引入“比例”项往往是不够的，比例项的作用仅是放大误差的幅值，因而需要增加的是“微分项”，它能预测误差变化的趋势，这样具有比例+微分的控制器就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免被控量的严重超调。所以对有较大惯性或滞后的被控对象，比例+微分（PD）控制器能改善系统在调节过程中的动态特性。

4．闭环控制系统特点

控制系统一般包括开环控制系统和闭环控制系统。开环控制系统（Open-loop Control System）是指被控对象的输出（被控制量）对控制器（controller）的输出没有影响。在这种控制系统中，不依赖将被控制量返送回来以形成任何闭环回路。闭环控制系统（Closed-loop Control System）的特点是：系统被控对象的输出（被控制量）会反送回来影响控制器的输出，形成一个或多个闭环。闭环控制系统有正反馈和负反馈，若反馈信号与系统给定值信号相反，则称为负反馈（Negative Feedback）；若极性相同，则称为正反馈。一般闭环控制系统均采用负反馈，又称负反馈控制系统。可见，闭环控制系统性能远优于开环控制系统。

5．PID控制器的参数整定

PID 控制器的参数整定是控制系统设计的核心内容。它是根据被控过程的特性，确定 PID 控制器的比例系数、积分时间和微分时间的大小。PID 控制器参数整定的方法很多，概括起来有如下两大类：

1）理论计算整定法。它主要依据系统的数学模型，经过理论计算确定控制器参数。这种方法所得到的计算数据未必可以直接使用，还必须通过工程实际进行调整和修改。

2）工程整定法。它主要依赖于工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。PID 控制器参数的工程整定方法，主要有临界比例法、反应曲线法和衰减法。这 3 种方法各有其特点，其共同点都是通过试验，然后按照工程经验公式对控制器参数进行整定。但无论采用哪一种方法所得到的控制器参数，都需要在实际运行中进行最后的调整与完善。

现在一般采用的是临界比例法。利用该方法进行 PID 控制器参数的整定步骤如下：

1）首先预选择一个足够短的采样周期让系统工作。

2）仅加入比例控制环节，直到系统对输入的阶跃响应出现临界振荡，记下这时的比例放大系数和临界振荡周期。

3）在一定的控制度下通过公式计算得到 PID 控制器的参数。

6．PID控制器的主要优点

PID 控制器成为应用最广泛的控制器，它具有以下优点：

1）PID 算法蕴涵了动态控制过程中过去、现在、将来的主要信息，而且其配置几乎最优。其中，比例（P）代表了当前的信息，起纠正偏差的作用，使过程反应迅速。微分（D）在信号变化时有超前控制作用，代表将来的信息。在过程开始时强迫过程进行，过程结



束时减小超调，克服振荡，提高系统的稳定性，加快系统的过渡过程。积分（I）代表了过去积累的信息，它能消除静差，改善系统的静态特性。此 3 种作用配合得当，可使动态过程快速、平稳、准确，得到良好的效果。

2）PID 控制适应性好，有较强的鲁棒性，对各种工业场合，都可在不同的程度上应用。特别适于“一阶惯性环节+纯滞后”和“二阶惯性环节+纯滞后”的过程控制对象。

3）PID 算法简单明了，各个控制参数相对较为独立，参数的选定较为简单，形成了完整的设计和参数调整方法，很容易为工程技术人员所掌握。

4）PID 控制根据不同的要求，针对自身的缺陷进行了不少改进，形成了一系列改进的 PID 算法。例如，为了克服微分带来的高频干扰的滤波 PID 控制；为克服大偏差时出现饱和超调的 PID 积分分离控制；为补偿控制对象非线性因素的可变增益 PID 控制等。这些改进算法在一些应用场合取得了很好的效果。同时当今智能控制理论的发展，又形成了许多智能 PID 控制方法。

7．PID算法

PID 控制器调节输出，保证偏差（e）为零，使系统达到稳定状态，偏差是给定值（SP）和过程变量（PV）的差。PID 控制的原理基于以下公式：

$$M(t) = K_C \cdot e + K_C \int_0^1 e\mathrm{d}t + M_{\text{initial}} + K_C \cdot \frac{\mathrm{d}e}{\mathrm{d}t}$$

公式中，$M(t)$ 是 PID 回路的输出；K_C 是 PID 回路的增益；e 是 PID 回路的偏差（给定值与过程变量的差）；M_{initial} 是 PID 回路输出的初始值。

由于以上的算式是连续量，必须将连续量离散化才能在计算机中运算，离散处理后的算式如下：

$$M_n = K_C \cdot e_n + K_I \cdot \sum_1^n e_x + M_{\text{initial}} + K_D \cdot (e_n - e_{n-1})$$

公式中，M_n 是在采样时刻 n，PID 回路的输出的计算值；K_C 是 PID 回路的增益；K_I 是积分项的比例常数；K_D 是微分项的比例常数；e_n 是采样时刻 n 的回路偏差值；e_{n-1} 是采样时刻 n－1 的回路偏差值；e_x 是采样时刻 x 的回路偏差值；M_{initial} 是 PID 回路输出的初始值。

再对以上算式进行改进和简化，得出如下计算 PID 输出的算式：

$$M_n = MP_n + MI_n + MD_n$$

公式中，M_n 是第 n 采样时刻的计算值；MP_n 是第 n 采样时刻的比例项值；MI_n 是第 n 采样时刻的积分项的值；MD_n 是第 n 采样时刻微分项的值。

7.2　利用S7-300 PLC进行电炉的温度控制

S7-300 PLC 提供有 PID 控制功能块来实现 PID 控制。STEP 7 提供了系统功能块 SFB41、SFB42、SFB43 实现 PID 闭环控制，其中 SFB41“CONT_C”用于连续控制，SFB42

“CONT_S”用于步进控制，SFB43“PULSEGEN”用于脉冲宽度调制，它们位于文件夹“Libraries→Standard Library→PID Controller”中。位于文件夹“Libraries→Standard Library→PID Controller”的 FB41、FB41、FB43 与 SFB41、SFB42、SFB43 兼容，FB58、FB59 则用于 PID 温度控制。它们是系统固化的纯软件控制器，运行过程中循环扫描、计算所需的全部数据存储在分配给 FB 或 SFB 的背景数据块里，因此可以无限次调用。

【例 7-1】 有一台电炉，要求炉温控制在一定的范围。电炉的工作原理如下：

当设定电炉温度后，CPU 314C-2DP 经过 PID 运算后由自带模拟量输出模块输出一个电压信号送到控制板，控制板根据电压信号（弱电信号）的大小控制电热丝的加热电压（强电）的大小（甚至断开），温度传感器测量电炉的温度，温度信号经过控制板的处理后输入到模拟量输入模块，再送到 CPU 314C-2DP 进行 PID 运算，如此循环。整个系统的硬件配置如图 7-1 所示。请编写控制程序。

【解】

1．主要软硬件配置

①1 套 STEP 7　V5.4SP4。

②1 台 CPU 314C-2DP。

③1 根编程电缆（或者 CP5611 卡）。

④1 台电炉（含控制板）。

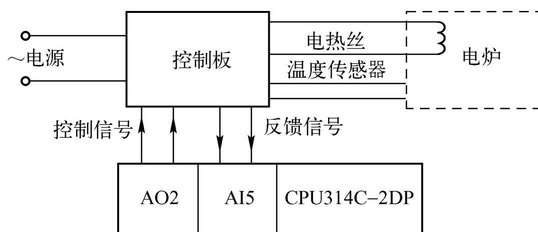


图 7-1　硬件配置图

2．硬件组态

1）新建工程，并插入站点。新建工程，命名为“PID（FB41）”，插入站点“SIMATIC 300（1）”，再在“Blocks”里插入组织块“OB35”、“OB100”和参数表“VAT_1”，如图 7-2 所示。

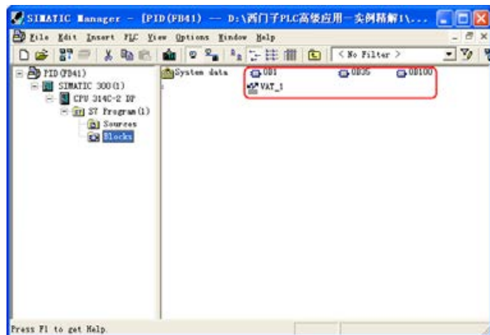


图 7-2　新建工程并插入站点



2）更改地址。双击“1”处的“DI24/DO16”，将数字量输入/输出的起始地址修改成从 0 开始，双击“1”处的“AI5/AO2”，将模拟量输入/输出的起始地址修改成从 3 开始，如图 7-3 所示。

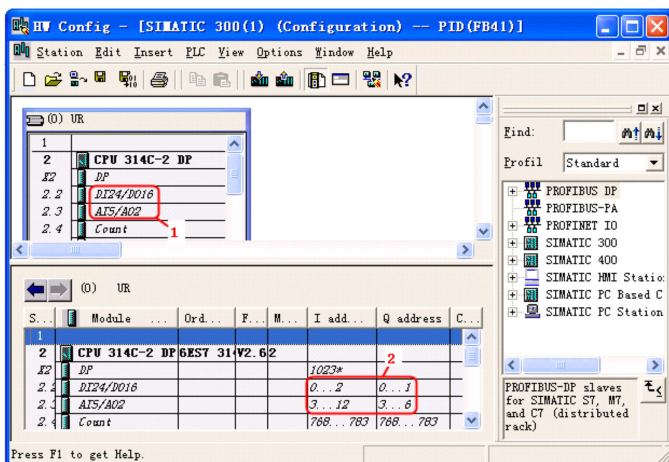


图 7-3　更改地址

3）设置模拟输入量测量范围。先选定“输入”选项卡，再选定“温度单位”为“摄氏度”，选择测量范围为“0～10V”，最后单击“确定”按钮，如图 7-4 所示。

4）设置模拟输出量测量范围。先选定“输出”选项卡，再选择输出电压范围为“0～10V”，单击“确定”按钮，如图 7-5 所示。

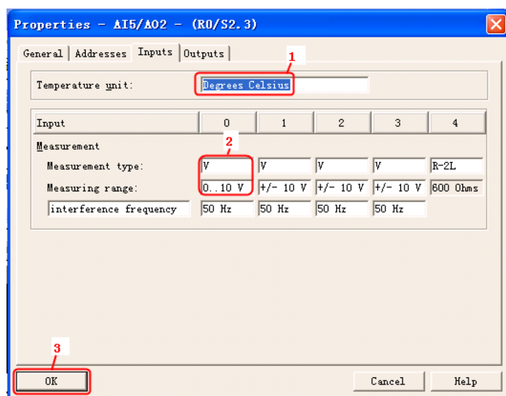


图 7-4　设置模拟输入量测量范围

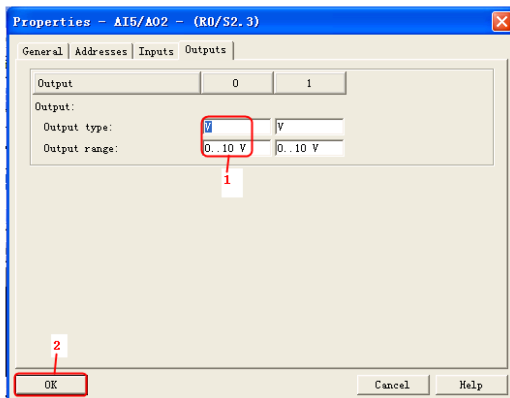


图 7-5　设置模拟输出量测量范围

3. 相关指令介绍

（1）SFB41 指令

SFB41 “CONT_C”（连续控制器）在 SIMATIC S7 可编程序逻辑控制器上使用，通过持续的输入和输出变量来控制工艺过程。在参数分配期间，可以通过激活或取消激活 PID 控制器的子功能使控制器适应工艺过程的需要。

可以使用该控制器作为 PID 固定设定值控制器或在多循环控制中作为层叠、混料或比率

控制器。该控制器的功能基于使用模拟信号的采样控制器的 PID 控制算法，必要时可以通过加入脉冲发生器阶段进行扩展，为使用成比例执行机构的两个或三个步骤控制器生成脉冲持续时间调制输出信号。但要注意只有在以固定时间间隔调用块时，在控制块中计算的值才是正确的。为此，应该在周期性中断 OB（OB30～OB38）中调用控制块。在 CYCLE 参数中输入采样时间。SFB41 指令的主要参数见表 7-1。

表 7-1　SFB41 指令参数

LAD	输入/输出	含义	数据类型
SFB41 EN　ENO COM_RST　LMN MAN_ON　LMN_PER PVPER_ON　QLMN_HLM P_SEL　QLMN_LLM I_SEL　LMN_P INT_HOLD　LMN_I I_ITL_ON　LMN_D D_SEL　PV CYCLE　ER SP_INT PV_IN PV_PER MAN GAIN TI TD TM_LAG DEADB_W LMN_HLM LMN_LLM PV_FAC PV_OFF LMN_FAC LMN_OFF I_ITLVAL DISV	EN	使能	BOOL
	COM_RST	为 1 时，重新启动 PID，复位 PID 内部参数	BOOL
	MAN_ON	为 1 时控制循环中断，直接将 MAN 的值送到 LMN	BOOL
	PVPER_PV	为 1 时，使用 I/O 输入的过程变量	BOOL
	P_SEL	为 1 时，打开比例 P 操作	BOOL
	I_SEL	为 1 时，打开积分 I 操作	BOOL
	NT_HOLD	为 1 时，积分输出被冻结	BOOL
	I_ITL_ON	为 1 时，I_ITLVAL 作为积分初值	BOOL
	D_SEL	为 1 时，打开微分 D 操作	BOOL
	CYCLE	采样时间	TIME
	SP_INT	PID 给定值	REAL
	PV_IN	浮点格式过程变量输入	REAL
	PV_PER	I/O 格式过程变量输入	WORD
	MAN	手动值	REAL
	GAIN	比例增益，用于设置控制器的增益	REAL
	TI	积分时间输入，积分响应时间	TIME
	TD	微分时间输入，微分响应时间	TIME
	TM_LAG	微分操作的延时时间输入	TIME
	DEADB_W	死去宽度	REAL
	LMN_HLM	控制器输出上限	REAL
	LMN_LLM	控制器输出下限	REAL
	PV_FAC	输入过程变量的比例因子	REAL
	PV_OFF	输入过程变量的偏移量	REAL
	LMN_FAC	输出过程变量的比例因子	REAL
	LMN_OFF	输出过程变量的偏移量	REAL
	I_ITLVAL	积分操作的初始值	REAL
	DISV	允许搅动量，一般不设置	REAL
	LMN	浮点格式的 PID 输出值	REAL
	LMN_PER	I/O 格式的 PID 输出值	WORD



（续）

LAD	输入/输出	含义	数 据 类 型
	QLMN_HLM	PID 输出值超出上限	BOOL
	QLMN_LLM	PID 输出值超出下限	BOOL
	LMN_P	PID 输出值中的比例成分	REAL
	LMN_I	PID 输出值中的积分成分	REAL
	LMN_D	PID 输出值中的微分成分	REAL
	PV	格式化的过程变量输出	REAL
	ER	死区处理后的误差输出	REAL

（2）FC105 指令

SCALE 功能接受一个整型值（IN），并将其转换为以工程单位表示的介于下限和上限（LO_LIM 和 HI_LIM）之间的实型值。将结果写入 OUT。SCALE 功能使用以下等式：

OUT = [((FLOAT (IN) −K1)/(K2−K1)) × (HI_LIM−LO_LIM)] + LO_LIM

常数 K1 和 K2 根据输入值是 BIPOLAR（双极性）还是 UNIPOLAR（单极性）设置。BIPOLAR 的含义是：假定输入整型值介于−27648 与 27648 之间，则 K1 = −27648.0，K2 = +27648.0。UNIPOLAR 的含义是：假定输入整型值介于 0 和 27648 之间，则 K1 = 0.0，K2 = +27648.0。

表 7-2　FC105 指令参数

LAD	输入/输出	含义	数据类型
"SCALE" EN　ENO IN　RET_VAL HI_LIM　OUT LO_LIM BIPOLAR	EN	使能，信号状态为 1 时激活该功能	BOOL
	HI_LIM	常数，工程单位表示的上限值	REAL
	LO_LIM	常数，工程单位表示的下限值	REAL
	BIPOLAR	为 1 时，表示输入值为双极性。信号状态 0 表示输入值为单极性	BOOL
	OUT	转换的结果	REAL
	RET_VAL	如果该指令的执行没有错误，将返回值 W#16#0000。对于 W#16#0000 以外的其他值，参见“错误信息”	WORD

4．编写程序

OB1 中的程序如图 7-6 所示；OB100 中的程序如图 7-7 所示，目的是重启 PID；OB35 中的程序如图 7-8 所示，每 0.1s 作一次 PID 运算。

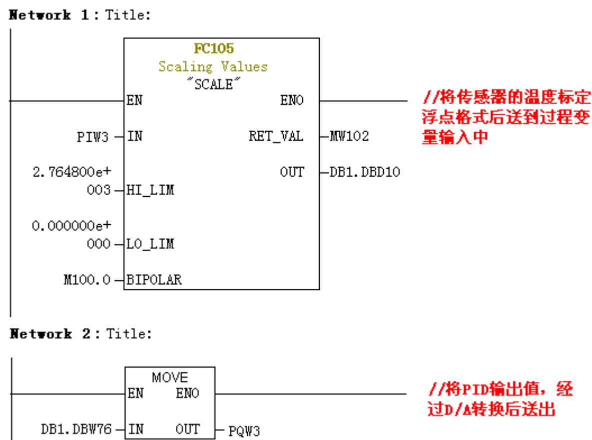


图 7-6　OB1 中的程序

```
Network 1 : Title:
    S       DB1.DBX     0.0         //重启PID
Network 2 : Title:
    R       DB1.DBX     0.0
```

图 7-7　OB100 中的程序

【关键点】S7-300 PLC 的程序中有的指令表是不能转换成梯形图的，因此在同一个程序中可能出现指令表和梯形图并存的情况，但这种情况在 S7-200 PLC 中是不会出现的。

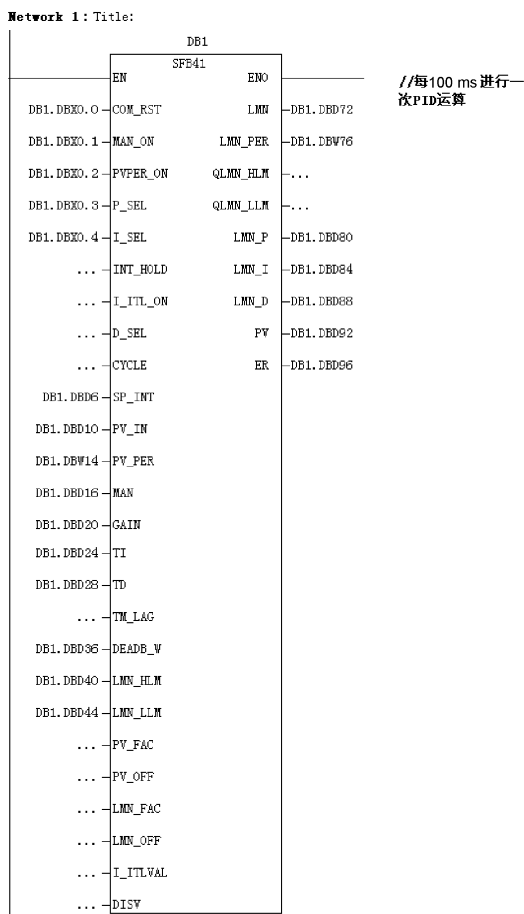


图 7-8　OB35 中的程序

小结

重点难点总结

1．对 PID 概念的理解和 PID 3 个参数的调节。

2．PID 指令的各参数含义。

习题

1．PID 3 个参数的含义是什么？



2．闭环控制有什么特点？

3．简述调整 PID 3 个参数的方法。

4．简述 PID 控制器的主要优点。

5．某水箱的出水口的流量是变化的，注水口的流量可通过调节水泵的转速控制，水位的检测可以通过水位传感器完成，水箱最大盛水高度为 2m，要求对水箱进行水位控制，保证水位高度为 1.6m。PLC 为控制器，用 SM331 为模拟量输入模块测量水位信号，用 EM332 输出信号控制变频器，从而控制水泵的输出流量。水箱的水位控制原理图如图 7-9 所示。

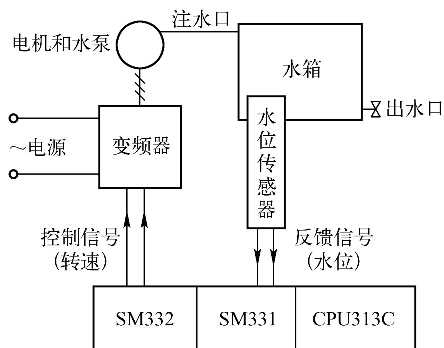


图 7-9　水箱的水位控制原理图

第 8 章

S7-300 PLC在模块化生产线变频调速系统中的应用

本章介绍 MM440 变频器的基本使用方法、PLC 控制变频器多段调速、PLC 控制变频器模拟量调速、PROFIBUS 现场总线通信调速。

8.1　认识模块化生产线的运输站

运输站比较简单，其结构简图如图 8−1 所示，由一台带减速机的三相异步交流电动机拖动滚筒旋转，减速器的减速比为 1∶15，传送带在滚筒的摩擦力的作用下输送工件，运输站的两端各有一个光电开关，光电开关 3 能够检测到上一站是否将工件送到运输站，而当光电开关 4 检测到有工件时，站上的气缸直接将工件推到下一站，其气动原理图如图 8−2 所示。为了节能，当光电开关 3 在 120s 内没有检测到工件，电动机停止转动。滚筒旋转速为 15～20r/min。已知电动机的技术参数，功率为 0.75kW，额定转速为 1440r/min，额定电压为 380V，额定电流为 2.05A，额定频率为 50Hz。

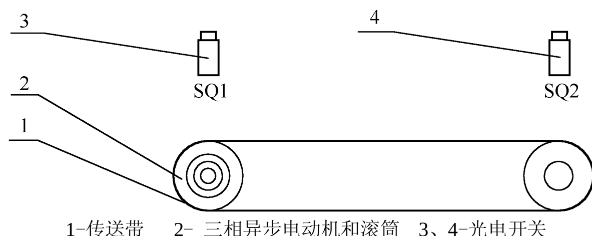


图 8−1　运输站简图

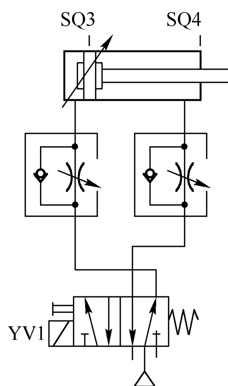


图 8−2　运输站气动原理图

8.2　西门子MM440 变频器使用简介

8.2.1　认识变频器

1．初识变频器

变频器一般是利用电力半导体器件的通断作用将工频电源变换为另一频率的电能控制装



置。变频器有着“现代工业维生素”之称，在节能方面的效果不容忽视。随着各界对变频器节能技术和应用等方面认识的逐渐加深，我国变频器市场变得异常活跃。

变频器产生的最初目的是速度控制，应用于印刷、电梯、纺织、机床和生产流水线等行业。而目前相当多的运用是以节能为目的。由于中国是能源消耗大国，而中国的能源储备又相对贫乏，因此国家大力提倡各种节能措施，其中着重推荐了变频器调速技术。在水泵和中央空调等领域，变频器可以取代传统的限流阀和回流旁路技术，充分发挥节能效果；在火电、冶金、矿山、建材行业，高压变频调速的交流电动机系统的经济价值正在得以体现。

变频器是一种高技术含量、高附加值、高效益回报的高科技产品，符合国家产业发展政策。在过去的 20 几年，我国变频器行业从起步阶段到目前正逐步开始趋于成熟，发展十分迅速。进入 21 世纪以来，我国中、低压变频器市场的增长速度超过了 20%，远远大于近几年的 GDP 增长水平。

从产品优势角度看，通过高质量地控制电动机转速，提高制造工艺水准，变频器不但有助于提高制造工艺水平（尤其在精细加工领域），而且可以有效节约电能，是目前最理想、最有前途的电动机节能设备。

从变频器行业所处的宏观环境看，无论是国家中长期规划、短期的重点工程、政策法规、国民经济整体运行趋势，还是人们节能环保意识的增强、技术的创新、发展高科技产业的要求，从国家相关部委到各相关行业，变频器都受到了广泛的关注，市场吸引力巨大。西门子变频器外形如图 8-3 所示。


图 8-3　变频器外形

2．交-直-交变频调速的原理

交-直-交变频调速的原理如图 8-4 所示，交-直-交变频调速就是变频器先将工频交流电整流成直流电，逆变器在微控制器（如 DSP）的控制下，将直流电逆变成不同频率的交流电。目前市面上的变频器多是这种工作原理。

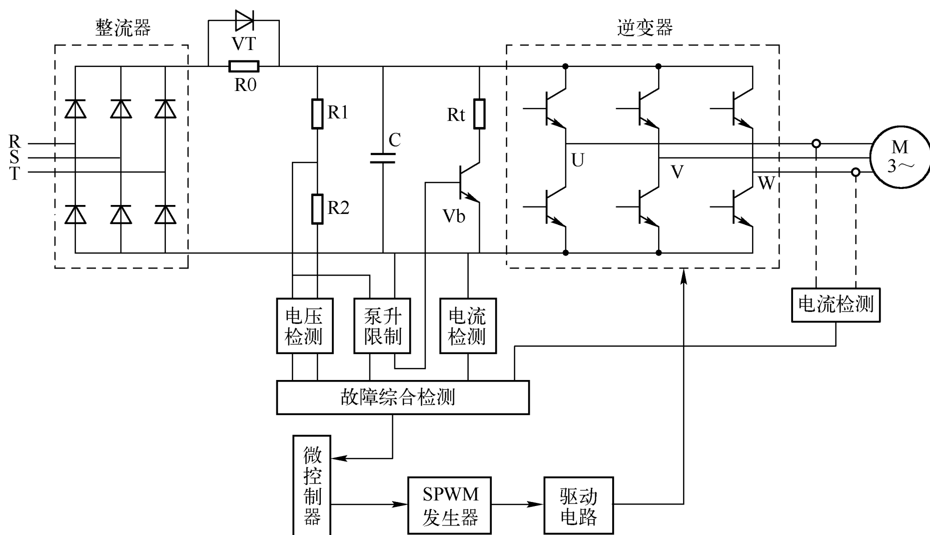


图 8-4　变频调速的原理图

图 8-4 中 R0 起限流作用，当 R、S、T 端子上的电源接通时，R0 接入电路，以限制起动电流。延时一段时间后，晶闸管 VT 导通，将 R0 短路，避免造成附加损耗。Rt 为能耗制动电阻，当制动时，异步电动机进入发动机状态，逆变器向电容 C 反向充电，当直流回路的电压，即电阻 R1、R2 上的电压升高到一定的值时（图中实际上测量的是电阻 R2 的电压），通过泵升电路使开关器件 Vb 导通，这样电容 C 上的电能就消耗在制动电阻 Rt 上。通常为了散热，制动电阻 Rt 安装在变频器外侧。电容 C 除了参与制动外，在电动机运行时，主要起滤波作用。顺便指出起滤波作用是电容器的变频器称为电压型变频器；起滤波作用是电感器的变频器称为电流型变频器，比较多见的是电压型变频器。

微控制器经运算输出控制正弦信号后，经过 SPWM（正弦脉宽调制）发生器调制，再由驱动电路放大信号，放大后的信号驱动 6 个功率晶体管，产生三相交流电压 U、V、W 驱动电动机运转。

【例 8-1】 如图 8-4 所示，若将变频器的动力线的输入和输出接反是否可行？若不可行有什么后果？

【解】

将变频器的动力线的输入和输出接反是不允许的，可能发生爆炸。

8.2.2　西门子MM440 变频器使用简介

1．初识西门子MM440 变频器

西门子 MM440 变频器由微处理器控制，并采用具有现代先进技术水平的绝缘栅双极性晶体管（IGBT）作为功率输出器件，它具有很高的运行可靠性和功能多样性。脉冲宽度调制的开关频率也是可选的，降低了电动机运行的噪声。

MM440 变频器的框图如图 8-5 所示，控制端子定义见表 8-1。

表 8-1　MM440 控制端子表

端子序号	端子名称	功能	端子序号	端子名称	功能
1	-	输出+10 V	16	DIN5	数字输入 5
2	-	输出 0 V	17	DIN6	数字输入 6
3	ADC1+	模拟输入 1（+）	18	DOUT1/NC	数字输出 1/常闭触点
4	ADC1-	模拟输入 1（-）	19	DOUT1/NO	数字输出 1/常开触点
5	DIN1	数字输入 1	20	DOUT1/COM	数字输出 1/转换触点
6	DIN2	数字输入 2	21	DOUT2/NO	数字输出 2/常开触点
7	DIN3	数字输入 3	22	DOUT2/COM	数字输出 2/转换触点
8	DIN4	数字输入 4	23	DOUT3NC	数字输出 3 常闭触点
9	-	隔离输出+24 V / max. 100 mA	24	DOUT3NO	数字输出 3 常开触点
10	ADC2+	模拟输入 2（+）	25	DOUT3COM	数字输出 3 转换触点
11	ADC2-	模拟输入 2（-）	26	DAC2+	模拟输出 2（+）
12	DAC1+	模拟输出 1（+）	27	DAC2-	模拟输出 2（-）
13	DAC1-	模拟输出 1（-）	28	-	隔离输出 0 V/max.100 mA
14	PTCA	连接 PTC/KTY84	29	P+	RS485
15	PTCB	连接 PTC/KTY84	30	P-	RS485



MM440 变频器的核心部件是 CPU 单元。根据设定的参数，经过运算，输出控制正弦波信号，正弦波信号再经过 SPWM 调制、放大，输出正弦交流电，驱动三相异步电动机运转。

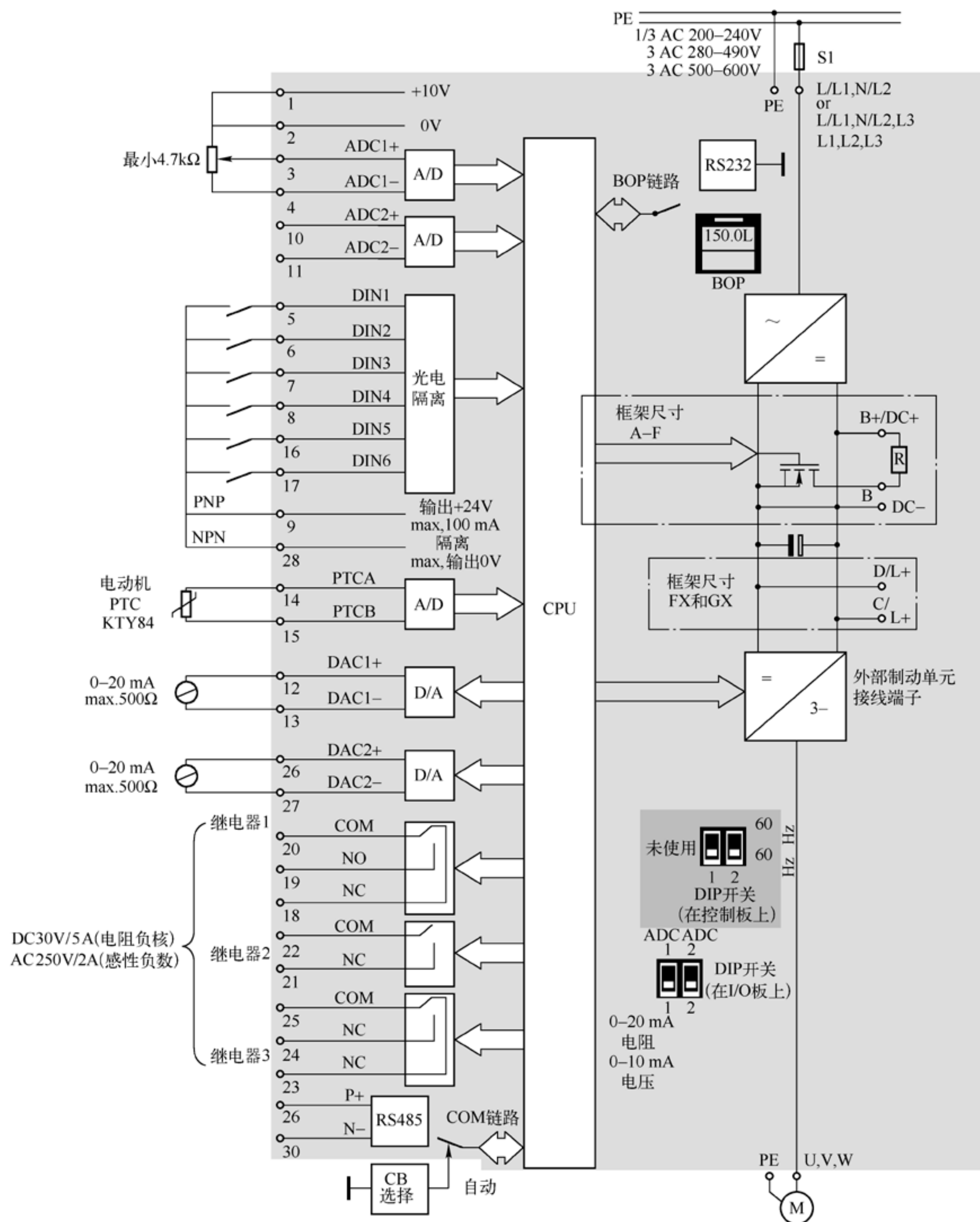


图 8-5　MM440 变频器的框图

MM440 变频器是一个智能化的数字变频器，在基本操作板上可进行参数设置，参数可

分为 4 个级别：

① 标准级，可以访问经常使用的参数。

② 扩展级，允许扩展访问参数范围，例如变频器的 I/O 功能。

③ 专家级，只供专家使用，即高级用户。

④ 维修级，只供授权的维修人员使用，具有密码保护。

【关键点】对于一般的用户，将变频器设置成标准级或者扩展级即可。

BOP 基本操作面板的外形如图 8-6 所示，利用基本操作面板可以改变变频器的参数。BOP 具有 7 段显示的 5 位数字，可以显示参数的序号和数值，报警和故障信息，以及设定值和实际值。参数的信息不能用 BOP 存储。BOP 基本操作面板上按钮的功能见表 8-2。



图 8-6　BOP 基本操作面板的外形

表 8-2　BOP 基本操作面板上的按钮的功能

显示/按钮	功能	功能的说明
r0000	状态显示	LED 显示变频器当前的设定值
I	启动变频器	按此按钮启动变频器。默认值运行时此按钮是被封锁的。为了使此按钮有效，应设定 P0700=1
0	停止变频器	OFF1：按此按钮，变频器将按选定的斜坡下降速率减速停车。默认值运行时此按钮被封锁，为了允许此按钮操作，应设定 P0700=1。OFF2：按此按钮两次（或一次，但时间较长），电动机将在惯性作用下自由停车，此功能总是“使能”的
	改变电动机的旋转方向	按此按钮可以改变电动机的旋转方向。电动机的反向用负号（－）表示或用闪烁的小数点表示。在默认设定时此按钮被封锁。为使此按钮有效，应先按“起动电动机”按钮
jog	电动机点动	在“准备合闸”状态下按压此按钮，则电动机起动并运行在预先设定的点动频率。当释放此按钮，电动机停车。当电动机正在旋转时，此按钮无功能
Fn	功能	此按钮用于浏览辅助信息 变频器运行过程中，在显示任何一个参数时按下此按钮并保持不动 2s，将显示以下参数值（在变频器运行中，从任何一个参数开始）： 1．直流回路电压（用 d 表示，单位：V） 2．输出电流（A） 3．输出频率（Hz） 4．输出电压（用 o 表示， 单位为 V） 5．由 P0005 选定的数值（如果 P0005 选择显示上述参数中的任何一个（3、4 或 5），这里将不再显示） 连续多次按下此按钮，将轮流显示以上参数 跳转功能，在显示任何一个参数（rXXXX 或 PXXXX）时，短时间按下此按钮，将立即跳转到 r0000，如果需要的话，可以接着修改其他的参数。跳转到 r0000 后，按此按钮将返回原来的显示点
P	访问参数	按此按钮即可访问参数
	增加数值	按此按钮即可增加面板上显示的参数数值



（续）

显示/按钮	功能	功能的说明
	减少数值	按此按钮即可减少面板上显示的参数数值
Fn + P	AOP菜单	调出 AOP 菜单提示（仅用于 AOP）

2．MM440 变频器BOP调速

以下用一个例子介绍 MM440 变频器 BOP 调速的过程。

【例 8-2】 一台 MM440 变频器配一台西门子三相异步电动机，已知电动机的技术参数，功率为 0.75kW，额定转速为 1380r/min，额定电压为 380V，额定电流为 2.05A，额定频率为 50Hz，试用 BOP 设定电动机的运行频率设定为 10Hz。

【解】

1）先介绍如何设定参数，以下通过将参数 P1000 的第 0 组参数，即设置 P1000[0]=1 的设置过程为例，讲解一个参数的设置方法。参数的设定方法见表 8-3。

表 8-3 参数的设定方法

序号	操作步骤	BOP显示
1	按P按钮，访问参数	r0000
2	按▲按钮，直到显示P1000	P1000
3	按P按钮，显示in000，即P1000的第0组值	in000
4	按P按钮，显示当前值2	2
5	按▼按钮，达到所要求的数值1	1
6	按P按钮，存储当前设置	P1000
7	按Fn按钮，显示r0000	r0000
8	按P按钮，显示频率	10.00

2）完整的设置过程。按照表 8-4 中的步骤进行设置。

表 8-4 设置过程

步骤	参数及设定值	说明	步骤	参数及设定值	说明
1	P0003=2	扩展级	9	P1080=0	最小频率
2	P0010=1	为 1 才能修改电动机参数	10	P1082=50	最大频率
3	P0304=380	额定电压	11	P1120=10	从静止到达最大频率所需时间
4	P0305=2.05	额定电流			
5	P0307=0.75	额定功率	12	P1121=10	从最大频率到停止所需时间
6	P0311=1380	额定转速			
7	P0700=1	命令源（起停）为 BOP	13	P0010=0	运行时必须为 0
8	P1000=1	频率源为 BOP			

3）起停控制。按下基本操作面板上的■按钮，三相异步电动机起动，稳定运行的频率为 10Hz；当按■按钮时，电动机停机。

【关键点】初学者在设置参数时，有时不注意进行了错误的设置，但又不知道在什么参数的设置上出错，通常这种情况下可以对变频器进行复位，一般的变频器都有这个功能，复位后变频器的所有参数变成出厂时设定值，但工程中正在使用的变频器要谨慎使用此功能。西门子 MM440 的复位方法是：先将 P0010 设置为 30，再将 P0970 设置为 1，变频器上的显示器中闪烁的“busy”消失后，变频器成功复位。

8.3　运输站的控制

由于运输站的变频调速控制方案比较简单，可以采用手动键盘调速（BOP）、模拟量调速、多段调速和通信调速，以下分别介绍。

8.3.1　运输站变频器的BOP调速

BOP 调速方式最为简单，消耗的资源最少，应优先采用。当采用 BOP 方式调速时，运输站的接线图如图 8-7 所示。

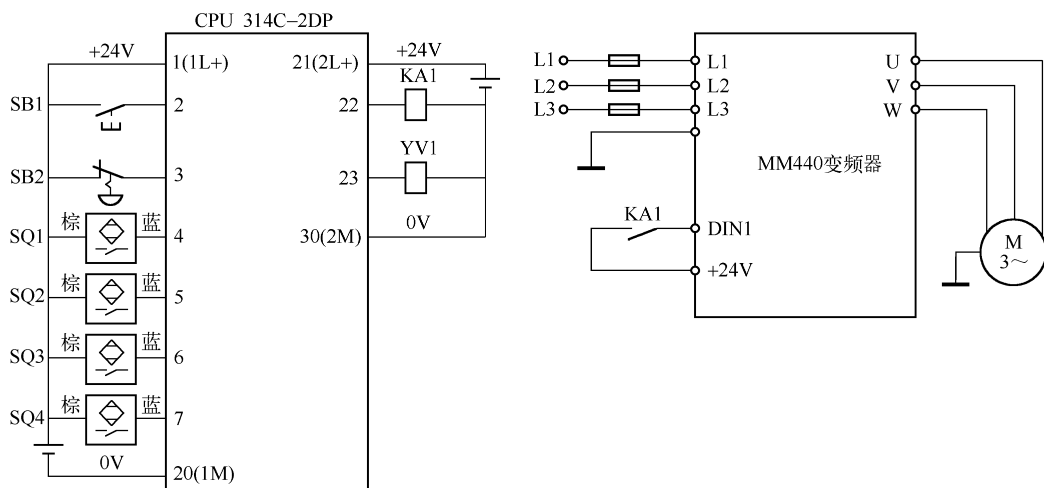


图 8-7　运输站接线图—BOP 调速方式

先要计算出电动机的输出频率，也就是在变频器中要设定的频率。因为减速器的传动比为 1:15，而要求最后减速器输出转速为 15～20 r/min。若减速器的输出转速为 16r/min，在要求的范围（15～20 r/min）中，则很容易算出电动机的输出转速为 $n = 16 \times i = 16 \times 15 = 240\text{r/min}$，又因为电动机的额定转速是 1440 r/min，其额定频率是 50 Hz，所以当电动机的转速为 240r/min 时，其频率为 $f = \frac{240}{1440} \times 50 = 8.6\text{Hz}$。变频器的参数设置见表 8-5。

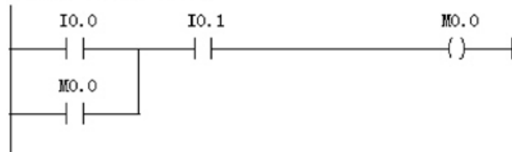


表 8-5　变频器参数表

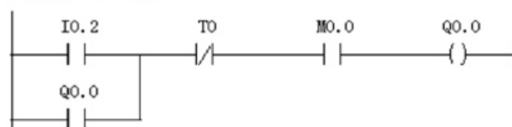
序号	变频器参数	出厂值	设定值	功能说明
1	P0304	380	380	电动机的额定电压（380V）
2	P0305	3.25	2.05	电动机的额定电流（2.05A）
3	P0307	0.75	0.75	电动机的额定功率（0.75kW）
4	P0310	50.00	50.00	电动机的额定频率（50Hz）
5	P0311	1440	1440	电动机的额定转速（1440 r/min）
6	P0700	2	2	选择命令源（由端子排输入）
7	P701	1	1	数字量输入 1
8	P1000	2	1	频率源（BOP）
9	P1001	5	8.6	固定频率值

编写运输站的程序控制程序如图 8-8 所示。

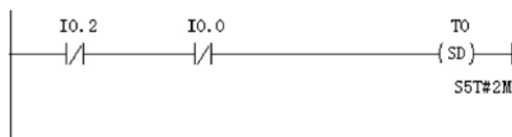
程序段 1：电动机起停



程序段 2：标题：



程序段 3：定时2分



程序段 4：气缸伸出和缩回

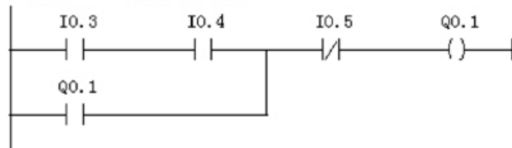


图 8-8　运输站程序

8.3.2　运输站变频器的模拟量调速

虽然 BOP 模式的调速简单易行，但每次改变频率需要手动设置，不易实现自动控制，而模拟量调速可以比较方便地实现自动控制和无级调速，因而在工程中比较常用，但模拟量调速一般要用到模拟量模块，相对而言，控制成本稍高。

1．S7-300 PLC模拟量模块简介

（1）S7-300 PLC 模拟量模块的分类

S7-300 PLC 模拟量模块的种类比较多，主要有模拟量输入模块 SM331 系列、模拟量输

出模块 SM332 系列以及模拟量输入/输出模块 SM334 和 SM335 系列，具体型号超过 30 种。按照分辨率分类，通常可分为 8 位、12 位和 16 位。

（2）接线方案

以下以 SM334（334-0CE01-0AA0）模块为例介绍模拟量模块的接线图。SM334 的接线图如图 8-9 所示。

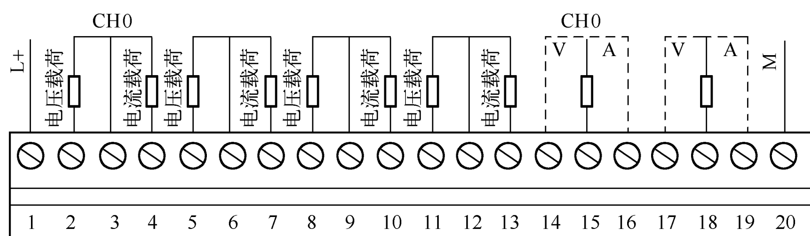


图 8-9　SM334 的接线图

2．运输站的控制方案和实施

（1）软硬件配置

① 1 套 STEP 7 V5.4SP4。

② 1 台 MM440 变频器。

③ 1 台 CPU 314C-2DP。

④ 1 台电动机。

⑤ 1 根编程电缆（或者 CP5611 卡）。

⑥ 1 台 SM334。

⑦ 1 台 HMI。

模拟量调速原理如图 8-10 所示。

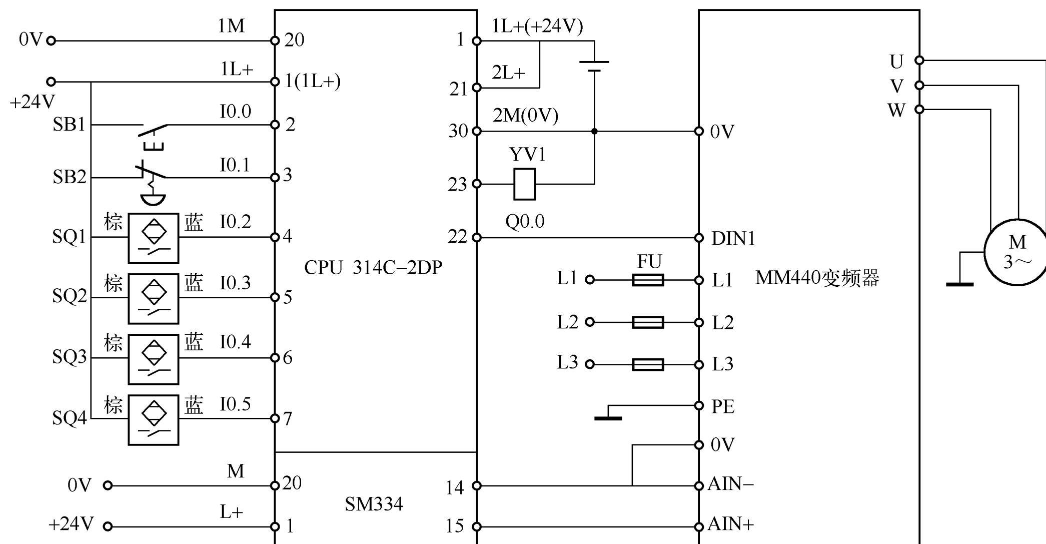


图 8-10　运输站接线图—模拟量调速



（2）设定变频器的参数

先查询 MM440 变频器的说明书，再依次在变频器中设定表 8−6 中的参数。

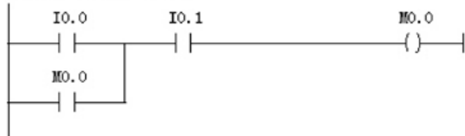
表 8-6　变频器参数表

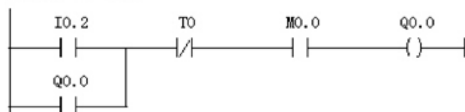
序号	变频器参数	出厂值	设定值	功能说明
1	P0304	380	380	电动机的额定电压（380V）
2	P0305	3.25	2.05	电动机的额定电流（2.05A）
3	P0307	0.75	0.75	电动机的额定功率（0.75kW）
4	P0310	50.00	50.00	电动机的额定频率（50Hz）
5	P0311	1440	1440	电动机的额定转速（1440 r/min）
6	P0700	2	2	选择命令源（由端子排输入）
7	P0756	0	0	选择 ADC 的类型（电压信号）
8	P1000	2	2	频率源（模拟量）
9	P701	1	1	数字量输入 1

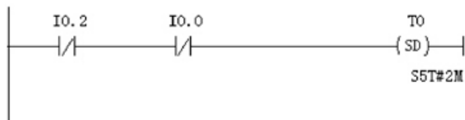
【关键点】P0756 设定成 0 表示电压信号对变频器调速，默认是电压信号。此外，还要将 I/0 控制板上的 DIP 开关设定为“OFF”。

（3）编写程序，并将程序下载到 PLC 中

在组织块 OB1 输入如图 8−11 程序所示的梯形图，其功能是发送起停信号和将物体推到下一站。在组织块 OB35 输入如图 8−12 程序所示的梯形图，其功能是每 100ms 向变频器发送一个模拟量，其中 MW20 中的数据由触摸屏提供。







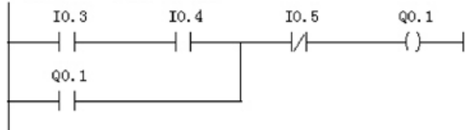


图 8−11　程序（组织块 OB1）

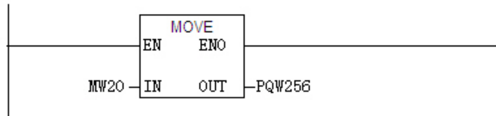


图 8−12　程序（组织块 OB35）

8.3.3　运输站变频器的多段调速

基本操作面板进行手动调速方法简单，对资源消耗少，但这种调速方法对于操作者来说比较麻烦，而且不容易实现自动控制，而 PLC 控制的多段调速和通信调速，就容易实现自

动控制，以下将介绍 PLC 控制的多段调速。

1．主要软硬件配置。

① 1 套 STEP 7 V5.4SP4。

② 1 台 MM440 变频器。

③ 1 台 CPU 314C−2DP。

④ 1 台电动机。

⑤ 1 根编程电缆（或者 CP5611 卡）。

运输站的接线图如图 8−13 所示。

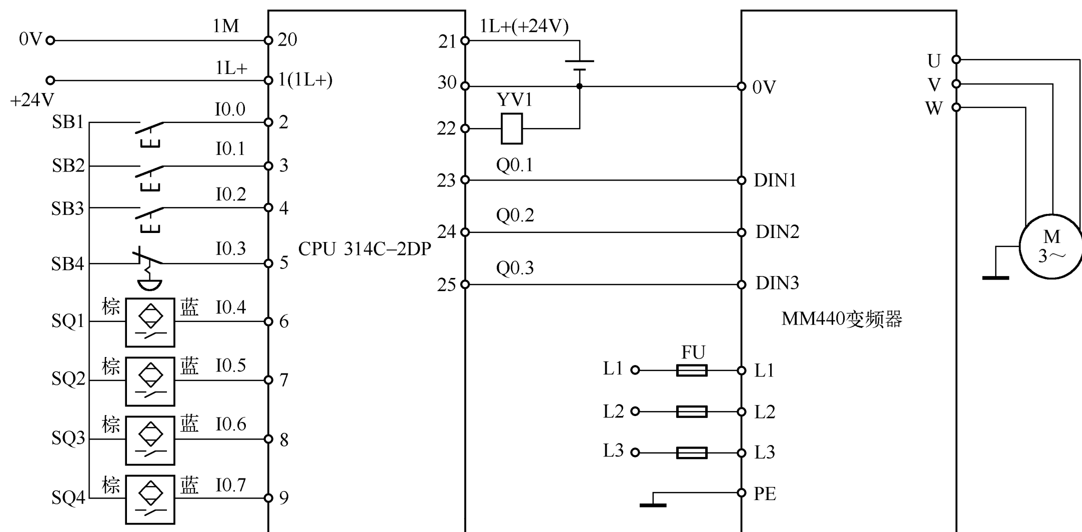


图 8−13　运输站的接线图－多段调速

2．参数的设置

多段调速时，当 DIN1 端子与变频器的 24V（端子 9）连接（或者与之相连 PLC 的输出点为高电平）时对应一个频率， DIN2 端子与变频器的 24V（端子 9）连接时再对应一个频率，DIN3 端子与变频器的 24V 接通时再对应一个频率。变频器参数见表 8−7。

表 8−7　变频器参数

序号	变频器参数	出厂值	设定值	功能说明
1	P0304	380	380	电动机的额定电压（380V）
2	P0305	3.25	2.05	电动机的额定电流（2.05A）
3	P0307	0.75	0.75	电动机的额定功率（0.75kW）
4	P0310	50.00	50.00	电动机的额定频率（50Hz）
5	P0311	1440	1440	电动机的额定转速（1440 r/min）
6	P1000	2	3	固定频率设定
7	P1080	0	0	电动机的最小频率（0Hz）
8	P1082	50	50.00	电动机的最大频率（50Hz）
9	P1120	10	10	斜坡上升时间（10s）
10	P1121	10	10	斜坡下降时间（10s）
11	P0700	2	2	选择命令源（由端子排输入）
12	P0701	1	16	固定频率设定值（直接选择+ON）



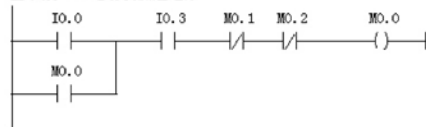
（续）

序号	变频器参数	出厂值	设定值	功能说明
13	P0702	12	16	固定频率设定值（直接选择+ON）
14	P0703	9	16	固定频率设定值（直接选择+ON）
15	P1001	0.00	5	固定频率 1=5Hz
16	P1002	5.00	10	固定频率 2=10Hz
17	P1003	10.00	20	固定频率 3=20Hz

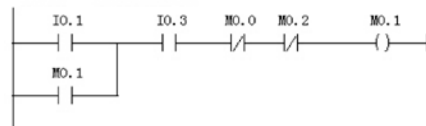
3．编写程序

这个程序相对比较简单，如图 8-14 所示。

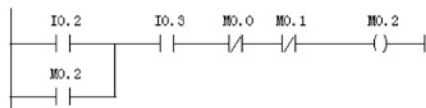
程序段 1：电动机低速起停



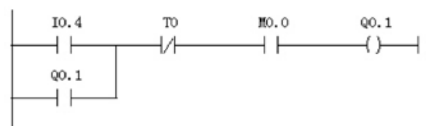
程序段 2：电动机中速起停



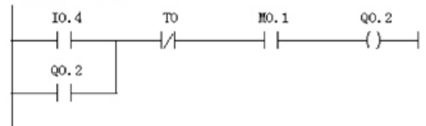
程序段 3：电动机高速起停



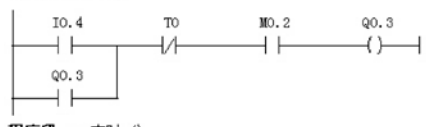
程序段 4：标题：



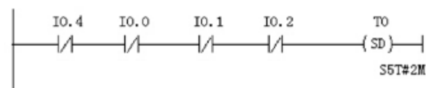
程序段 5：标题：



程序段 6：标题：



程序段 7：定时2分



程序段 8：气缸伸出和缩回

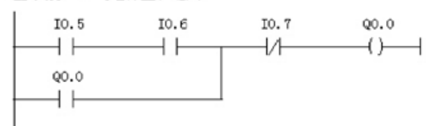


图 8-14　程序

8.3.4　运输站变频器的通信调速

通信调速既可实现无级调速，也可实现自动控制，应用灵活方便。S7−300/400 PLC 与 MM440 变频器可进行 USS 和 PROFIBUS 通信。MM440 变频器也只能作 PROFIBUS−DP 从站，不能作 PROFIBUS−DP 主站，而且 MM440 变频器用做从站时要配置通信模板。

1．MM440 变频器通信的基本知识

MM440 变频器既支持和主站的周期性数据通信，也支持和主站的非周期性数据通信，也就是说 S7−300 PLC 可以使用功能块 SFC14/SFC15 读取和修改 MM440 的参数值，调用一次可以读取或者修改一个参数。同时也可以使用功能块 SFC58/SFC59 或者 SFB52/SFB53 读取或者修改 MM440 的参数，一次最多可以读取或者修改 39 个参数。

有效的数据块分成两个区域即 PKW 区（参数识别 ID-数值区）和 PZD 区（过程数据），有效数据字符如图 8−15 所示。

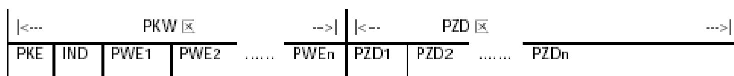


图 8−15　有效数据字符

PKW 区说明（参数识别 ID-数值区）PKW 接口的处理方式。PKW 接口并非物理意义上的接口，而是一种机理。这一机理确定了参数在两个通信伙伴之间（例如控制装置与变频器）的传输方式，例如参数数值的读和写。

PKW 区的结构。PKW 区前两个字（即 PKE 和 IND）的信息是关于主站请求的任务识别标记 ID 或应答报文的类型（应答识别标记 ID）PKW 区的第三、第四个字，规定报文中要访问的变频器参数号（PNU）的编号与 MICROMASTER4 的参数号相对应例如 1082 = P1082 = Fmax。第一个字 PKE 见表 8−8，第二个字 IND 见表 8−9。

表 8−8　第一个字 PKE

第一个字16 位=PKE=参数识别标记ID		
位18～12	AK =任务或应答识别标记ID	参看下文
位11	SPM =参数修改报告	不支持（总是0）
位10～00	b.PNU =基本参数号	完整的PNU由基本参数号与IND的18～12一起构成

表 8−9　第二个字 IND

第二个字16 位=IND=参数的下标		
位15141312（$2^0 2^3 2^2 2^1$）	PNU 扩展PNU 页号	参看下文
位11～10	备用	未使用
位09～08	选择文本的类型+文本的读或写	未使用
位07～00	下标哪个元素 哪个参数值 哪个元素说明 哪个下标文本是有效的 哪个数值文本是有效的	数值255＝下表参数数值或参数说明的全部元素。只有当P2013=127 时才有可能

完整的参数号是由参数的任务/应答识别 ID（位 0−10）中的基本参数号和下标 PNU 页号中的位 12～15 一起产生的。第二个字 IND 参数下标见表 8−10。



表 8-10　第二个字 IND 参数下标

基本参数号任务/应答识别标记ID中的位10～0	PNU 页（下标中的位18-12）	完整的PNU =基本PNU+ PNU （页号*2000）
0…1999	0	0…1999
0…1999	1	2000…3999
0…1999	2	4000…5999
…	…	…
0…1999	15	30000…31999

第三个字 PWE1 和第四个字 PWE2 是被访问参数的数值，见表 8-11 和表 8-12。MICROMASTER4 的参数数值有许多不同的类型：整数（单字长或双字长）、十进制数（以 IEEE 浮点数的形式给出永远是双字长）以及下标参数（这里称为数组）。参数的含义决定于参数数值的类型和 P2013 的设置。

表 8-11　第三个字 PWE1

第三个字PWE1=第一个参数数值		
	参数数值的类型	P2013 的设置
位18～0	=对于非数组参数是参数的数值 =对于数组参数是第n个参数的数值和对于第n个元素的任务	当P2013=3（ 固定长度为3 个字）或=127（长度可变）以及单字长参数时
	=对于数组参数是第1个参数的数值和对于所有元素的任务	当P2013 =127（长度可变）以及单字长参数时
	=0	当P2013 =4（固定长度为4 个字）以及单字长参数时
	=参数数值的高位字（非数组参数） =对于数组参数是参数数值的高位字和对于第n个元素任务的高位字	当P2013=4（ 固定长度为4 个字）或=127（长度可变）以及双字长参数时
	=对于数组参数是第一个参数数值的高位字和对于所有元素任务的高位字	当P2013 =127（长度可变）以及双字长参数时
	错误的数值	从站向主站发送，且应答标记ID=任务不能执行时

表 8-12　第四个字 PWE2

第四个字PWE2=第二个参数数值		
	参数数值的类型	P2013 的设置
位18～0	=对于数组参数是第2个参数数值和对于所有元素的任务	当P2013=4（ 固定长度为4 个字）或=127（长度可变）以及单字长参数时
	=参数数值的低位字非数组参数 =对于数组参数是第n个参数数值的低位字和对于第n个元素任务的低位字	当P2013=4（ 固定长度为4 个字）或=127（长度可变）以及双字长参数时
	=对于数组参数是第1个参数数值的低位字和对于所有元素任务的低位字	当P2013=127（长度可变）以及双字长参数时
	=下一个要访问的识别符标记ID	从站向主站传送，且应答识别标记ID=任务不能执行时。错误的数值=ID 不存在或ID不能访问时 当P2013=127长度可变时
	=下一个或前一个有效的数值16位 =下一个或前一个有效的数值32位的高位字 根据以下判定条件： 如果新值>实际值 向下一个有效的数值 如果新值<实际值向前一个有效的数值	从站向主站传送且应答识别标记ID=任务不能执行时。错误的数值=数值不可接受或有新的最大/最小值存在。当P2013=127 长度可变时

参数识别标记 ID（PEK）总是一个 16 位的值，位 0～10（PNU）包括所请求的参数号码，位 11（SPM）用于参数变更报告的触发位，位 12～15（AK）包括任务识别标记 ID。见表 8-13，应答标记 ID 见表 8-14。

表 8-13　任务识别标记 ID

任务识别标记 ID	含义	识别标记 ID	
		正	负
0	没有任务	0	-
1	请求参数数值	1 或 2	7
2	修改参数数值（单字）[只是修改 RAM]	1	7 或 8
3	修改参数数值（双字）[只是修改 RAM]	2	7 或 8
4	请求元素说明	3	7
5	修改元素说明（MICROMASTER4 不可能）	-	-
6	请求参数数值（数组），即带下标的参数	4 或 5	7
7	修改参数数值（数组，单字）[只是修改 RAM]	4	7 或 8
8	修改参数数值（数组，双字）[只是修改 RAM]	5	7 或 8
9	请求数组元素的序号，即下标的序号，“no.”	6	7
10	保留备用	-	-
11	存储参数数值（数组，双字）[RAM 和 EEPROM 都修改]	5	7 或 8
12	存储参数数值（数组，单字）[RAM 和 EEPROM 都修改]	4	7 或 8
13	存储参数数值（双字）[RAM 和 EEPROM 都修改]	2	7 或 8
14	存储参数数值（单字）[RAM 和 EEPROM 都修改]	1	7 或 8
15	读出或修改文本（MICROMASTER440 不可能）	-	-

表 8-14　应答识别标记 ID

任务识别标记 ID	含义	对任务识别标记 ID
0	不应答	0
1	传送参数数值（单）字	1、2 或 14
2	传送参数数值（双字）	1、3 或 13
3	传送说明元素	4
4	传送参数数值（数组，单字）	6、7 或 12
5	传送参数数值（数组，双字）	6、8 或 11
6	传送数组元素的数目	9
7	任务不能执行（有错误的数值）	1～15
8	对参数接口没有修改权	2、3、5、7、8、11～14 或 15
9～12	未使用	-
13	预留，备用	-
14	预留，备用	-
15	传送文本	15

2．软硬件配置

① 1 套 STEP 7　V5.4SP4。



② 1 台 MM440 变频器（含 PROFIBUS 模板）。

③ 1 台 CPU 314C-2DP。

④ 1 台电动机。

⑤ 1 根编程电缆（或者 CP5611 卡）。

⑥ 1 根 PROFIBUS 屏蔽双绞线。

⑦ 1 台 HMI。

硬件配置如图 8-16 所示。

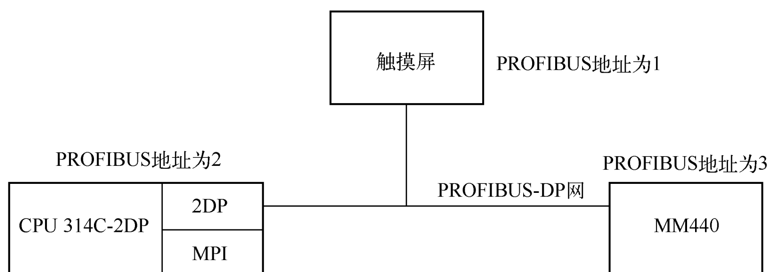


图 8-16　硬件配置图

将生产线的搬运站和运输站整体控制，其示意图如图 8-17 所示，两站的气动原理图如图 8-18 所示，I/O 接线图如图 8-19 所示。

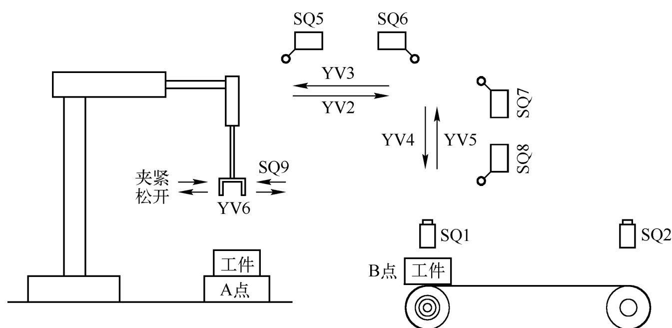


图 8-17　示意图

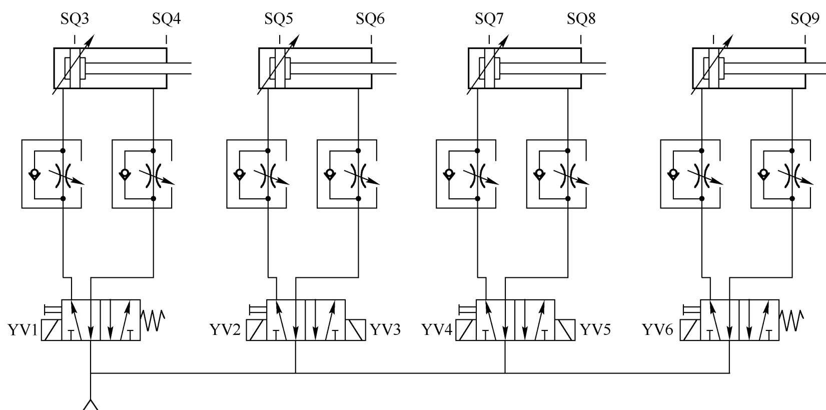


图 8-18　气动原理图

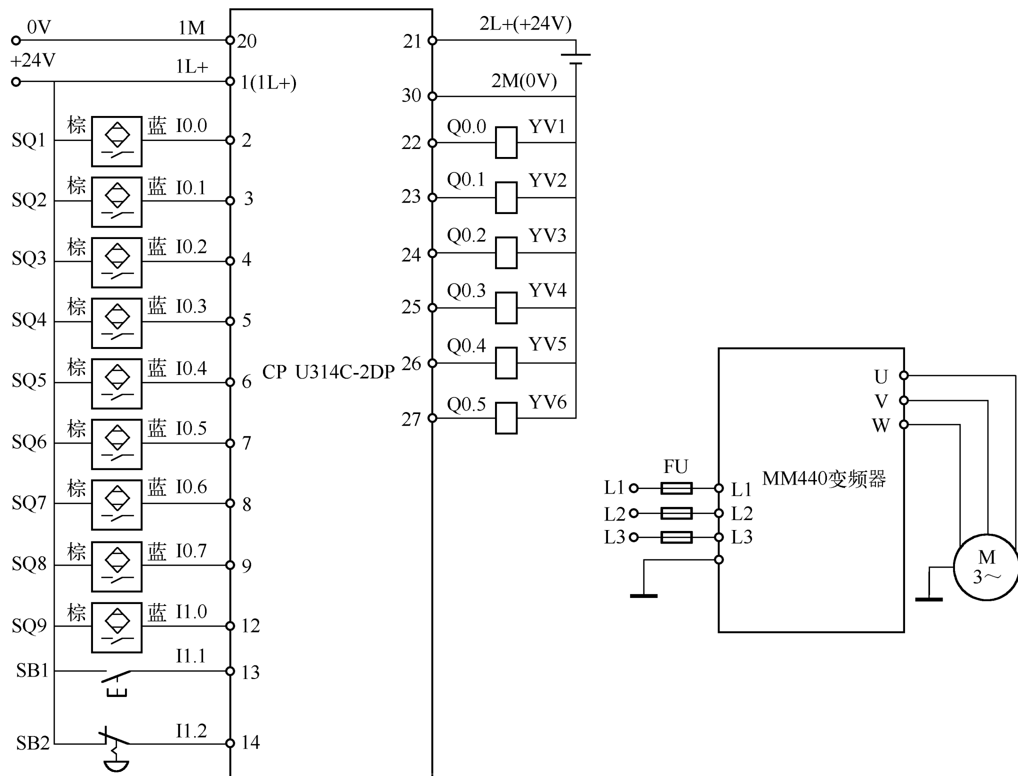


图 8-19　I/O 接线图

3．MM440 变频器的设置

MM440 变频器的参数见表 8-15。

表 8-15　变频器参数表

序号	变频器参数	出厂值	设定值	功能说明
1	P0304	380	380	电动机的额定电压（ 380V ）
2	P0305	3.25	2.05	电动机的额定电流（ 2.05A ）
3	P0307	0.75	0.75	电动机的额定功率（ 0.75kW ）
4	P0310	50.00	50.00	电动机的额定频率（ 50Hz ）
5	P0311	1440	1440	电动机的额定转速（ 1440 r/min ）
6	P0700	2	6	选择命令源（COM 链路的通信板 CB 设置 ）
7	P1000	2	6	频率源（COM 链路的通信板 CB 设置）
8	P2009	0	1	USS 规格化

MM440 变频器 PROFIBUS 站地址的设定在变频器的通信板（CB）上完成，通信板（CB）上有一排拨钮用于设置地址，每个拨钮对应一个“8-4-2-1”码的数据，所有的拨钮处于“ON”位置对应的数据相加的和就是站地址。拨钮示意图如图 8-20 所示，拨钮 1 和 2



处于“ON”位置，所以对应的数据为 1 和 2；而拨钮 3、4、5 和 6 处于“OFF”位置，所对应的数据为 0，站地址为 1+2+0+0+0+0+0=3。

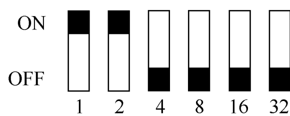


图 8-20　拨钮示意图

【关键点】图 8-20 设置的站地址为 3，必须和 STEP 7 软件中硬件组态的地址保持一致，否则不能通信。

4．S7-300 PLC的硬件组态

1）新建工程和 PROFIBUS 网络。将工程命名为“PROFIBUS”。新建“PROFIBUS”网络，设置“CPU 314C-2DP”的站地址为“2”，选中如图 8-21 所示中“1”处的网络，展开“PROFIBUS　DP”。

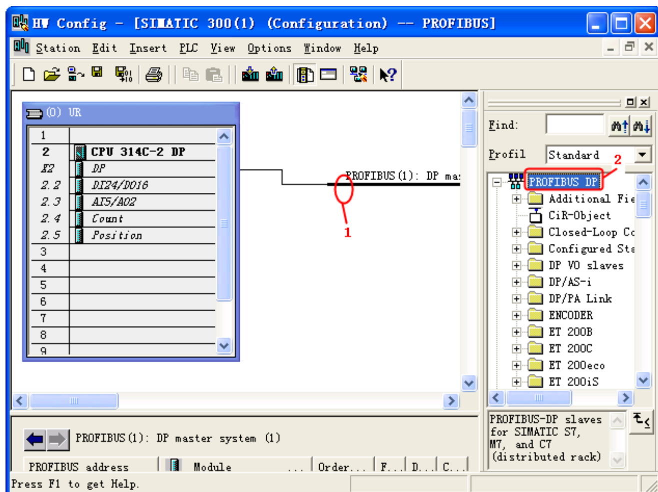


图 8-21　新建工程和 PROFIBUS 网络

2）选中“MICROMASTER 4”。如图 8-22 所示，先展开“SIMOVERT”，再选中“MICROMASTER 4”，并双击之，弹出 8-23 所示的界面。

3）设置 MM440 的站地址。如图 8-23 所示，先选中“PROFIBUS（1）”网络，再将“Address”（地址）设置为“3”，最后单击“OK”按钮。

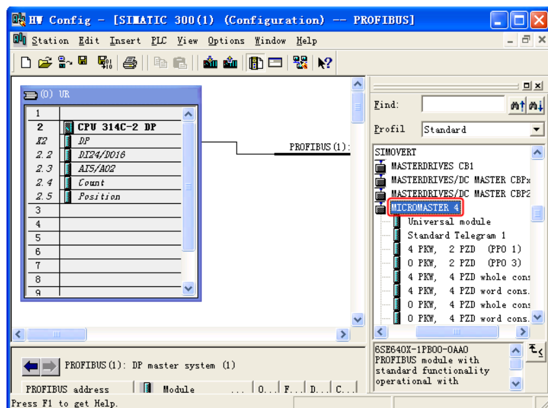


图 8-22　选中“MICROMASTER 4”

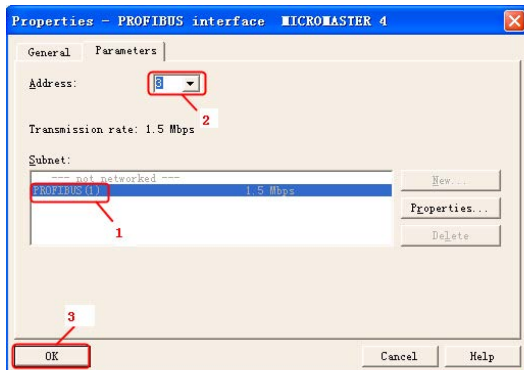


图 8-23　设置 MM440 的站地址

4）选择通信报文的结构。PROFIBUS 的通信报文由两部分组成，即 PKW（参数识别 ID 数据区）和 PZD 区（过程数据）。如图 8-24 所示，先选中“1”处，再双击“0 PKW ，2 PZD （PPO3)”，“0 PKW，2 PZD （PPO3)”通信报文格式的含义是：报文中没有 PKW，只有 2 个字的 PZD。

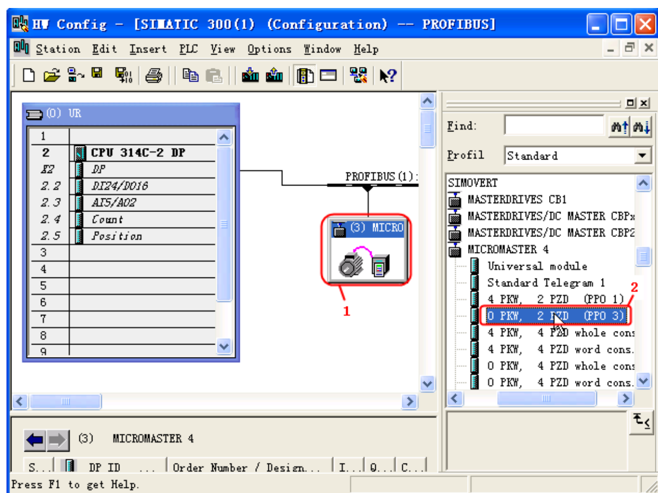


图 8-24　选择通信报文的结构

5）MM440 的数据地址。如图 8-25 中，MM440 接收主站的数据存放在 IB256~IB259（共两个字），MM440 发送信息给主站的数据区在 QB256~QB259（共两个字）。最后，编译并保存组态完成的硬件。

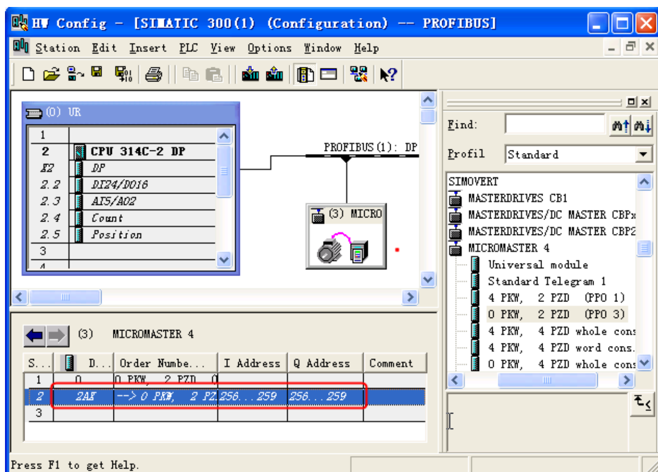


图 8-25　MM440 的数据地址

5. 编写程序

（1）任务报文 PZD 的介绍

任务报文的 PZD 区是为控制和检测变频器而设计的。PZD 的第一个字是变频器的控制字（STW）。变频器的 STW 控制字见表 8-16。



表 8-16　变频器的 STW 控制字

位	项目	含义
位 00	ON（斜坡上升）/OFF1（斜坡下降）	0 否，1 是
位 01	OFF2：按照惯性自由停车	0 是，1 否
位 02	OFF3：快速停车	0 是，1 否
位 03	脉冲使能	0 否，1 是
位 04	斜坡函数发生器（RFG）使能	0 否，1 是
位 05	RFG 开始	0 否，1 是
位 06	设定值使能	0 否，1 是
位 07	故障确认	0 否，1 是
位 08	正向点动	0 否，1 是
位 09	反向点动	0 否，1 是
位 10	由 PLC 进行控制	0 否，1 是
位 11	设定值反向	0 否，1 是
位 12	未使用	
位 13	用电动电位计 MOP 升速	0 否，1 是
位 14	用电动电位计 MOP 降速	0 否，1 是
位 15	本机/远程控制	0P7019 下标 0，1P7019 下标 1

PZD 的第二个字是变频器的主设定值（HSW），也是主频率设定值。有两种不同的设置方式，当 P2009 设置为 0 时，数值以十六进制形式发送，即 4000（hex）规格化为由 P2000（默认值为 50）设定的频率，4000 相当于 50Hz。当 P2009 设置为 1 时，数值以十进制形式发送，即 4000（十进制）表示频率为 40.00Hz。

例如当 P2009＝0 时，任务报文为 PZD＝047F4000，第一个字的二进制为 0000，0100，0111，1111。这个字的含义是：斜坡上升；不是自由惯性停机；不是快速停车；脉冲使能；斜坡函数发生器（RFG）使能；RFG 开始；设定值使能；不确认故障；不是正向点动；不是反向点动；PLC 进行控制；设定值不反向；不用 MOP 升速和降速。第二个字的含义是：转速为 50Hz。

（2）应答报文 PZD 的介绍

应答报文 PZD 的第一个字是变频器的状态字（ZWS）。变频器的状态字通常由参数 r0052 定义。变频器的状态字（ZSW）含义见表 8-17。

表 8-17　变频器的状态字 ZSW

位	项目	含义
位 00	变频器准备	0 否，1 是
位 01	变频器运行准备就绪	0 否，1 是
位 02	变频器正在运行	0 否，1 是
位 03	变频器故障	0 是，1 否
位 04	OFF2 命令激活	0 是，1 否
位 05	OFF3 命令激活	0 否，1 是
位 06	禁止接通	0 否，1 是
位 07	变频器报警	0 否，1 是
位 08	设定值/实际偏差过大	0 是，1 否

（续）

位	项目	含义
位 09	过程数据监控	0 否，1 是
位 10	已经达到最大频率	0 否，1 是
位 11	电动机极限电流报警	0 是，1 否
位 12	电动机抱闸制动投入	0 是，1 否
位 13	电动机过载	0 是，1 否
位 14	电动机正向运行	0 否，1 是
位 15	变频器过载	0 是，1 否

应答报文的 PZD 第二个字是变频器的运行实际参数（HIW）。通常定义为变频器的实际输出频率。其数值也由 P2009 进行规格化。

（3）编写程序

先根据题意画出功能图如图 8-26 所示，再编写程序如图 8-27、图 8-28 所示。

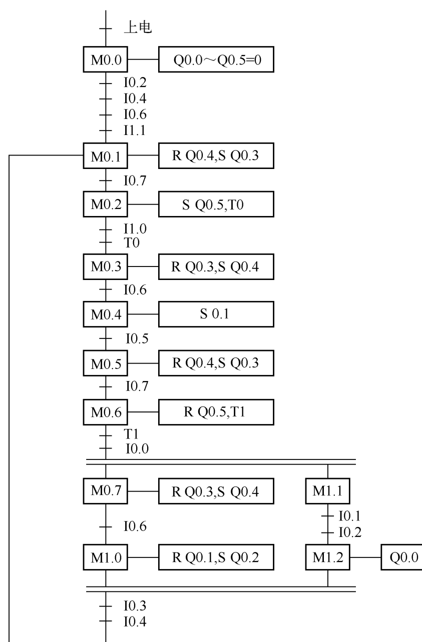
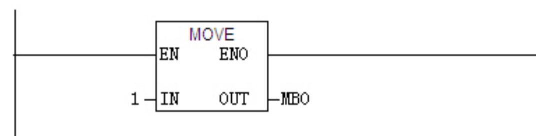


图 8-26　功能图



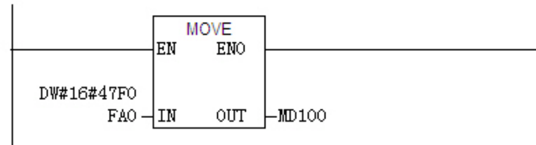
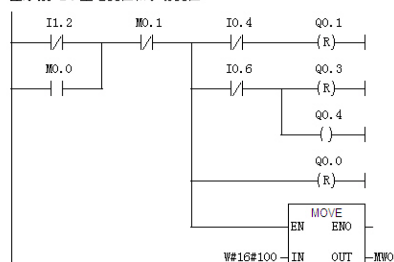


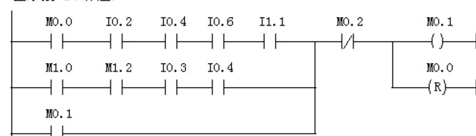
图 8-27　OB100 中的程序



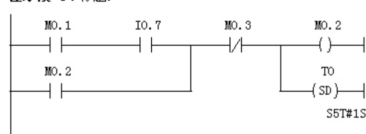
程序段 1：上电复位和手动复位



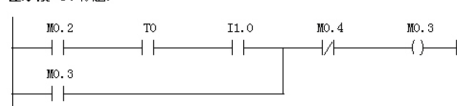
程序段 2：标题：



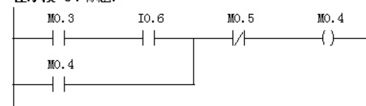
程序段 3：标题：



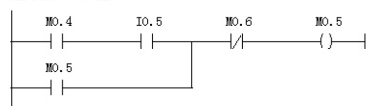
程序段 4：标题：



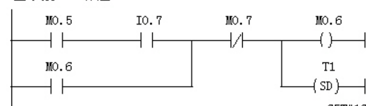
程序段 5：标题：



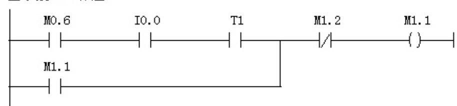
程序段 6：标题：



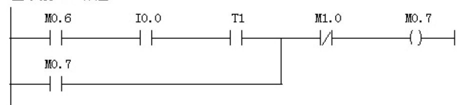
程序段 7：标题：



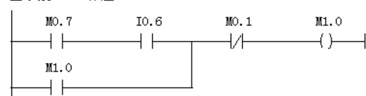
程序段 8：标题：



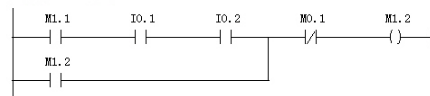
程序段 9：标题：



程序段 10：标题：



程序段 11：标题：



程序段 12：向上运行



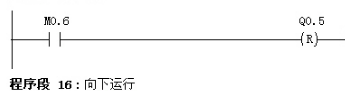
程序段 13：标题：



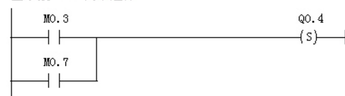
程序段 14：夹紧



程序段 15：松开



程序段 16：向下运行



程序段 17：标题：



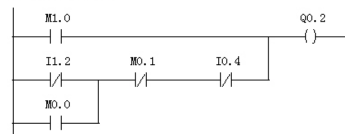
程序段 18：水平伸出



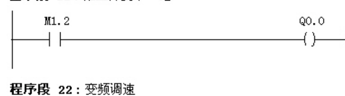
程序段 19：水平缩回



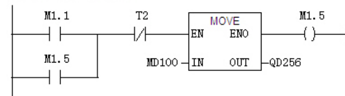
程序段 20：标题：



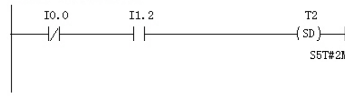
程序段 21：推工件到下一站



程序段 22：变频调速



程序段 23：定时2分钟



程序段 24：变频器控制电机停止

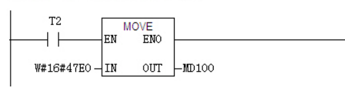


图 8-28　OB1 中的程序

【关键点】理解任务报文和应答报文各位的含义是十分关键的，否则是很难编写出正确程序。

小结

重点难点总结

1．掌握变频器的参数设定是正确使用变频器的前提。

2．理解变频器的“交－直－交”工作原理。

3．掌握变频器 BOP 调速、多段调速、模拟量调速和通信调速的应用场合以及其在运输站上的应用。

4．掌握 PLC 控制变频器调速的接线方法，特别注意当 PLC 为晶体管输出时，若 PLC 为 PNP 输出，则要将变频器的输入调整为 PNP 输入，同理若 PLC 为 NPN 输出，则要将变频器的输入调整为 NPN 输入。

5．通信调速的难点是理解各个控制字的含义，这是非常关键的，此外对变频器参数的正确设定也十分关键。

习题

1．简述变频器的“交－直－交”工作原理。

2．三相交流异步电动机有几种调速方式？

3．使用变频器时，一般有几种调速方式？

4．变频器电源输入端接到电源输出端后，有什么后果？

5．使用变频器时，制动原理是什么？

6．使用变频器时，电动机的正反转怎样实现？

第 9 章

上位机对S7-300/400 PLC的监控

本章介绍了组态软件的功能，以及怎样在上位机中用组态软件监控 S7-300/400 PLC。

9.1 简单组态软件工程的建立

9.1.1 认识组态软件

1．初识组态软件

在使用工控软件时，人们经常提到组态一词，组态的英文是“Configuration”，简而言之，组态就是利用应用软件中提供的工具、方法，完成工程中某一具体任务的过程。组态软件是数据采集监控系统 SCANDA（Supervisory Control and Data Acquisition）的软件平台工具，是工业应用软件的一个组成部分。它具有丰富的设置项目，使用方式灵活，功能强大。组态软件由早先的单一的人机界面向数据处理方向发展，管理的数据量越来越大。随着组态软件自身以及控制系统的发展，监控组态软件部分与硬件分离，为自动化软件的发展提供了充分发挥作用的舞台。OPC（OLE for Process Control）的出现，以及现场总线和工业以太网的快速发展，大大简化了不同厂家设备之间的互联，降低了开发 I/O 设备驱动软件的工作量。

实时数据库的作用进一步加强。实时数据库是 SCANDA 系统的核心技术。从软件技术上讲，SCANDA 系统的实时数据库，实际上就是一个可统一管理、支持变结构、支持实时计算的数据结构模型。

社会信息化的加速发展是组态软件市场增长的强大推动力。在最终用户眼里，组态软件在自动化系统中发挥的作用逐渐增大，甚至有时到了非用不可的地步。主要原因在于组态软件的功能强大，用户普遍需求，而且逐渐认识其价值。

2．组态软件的功能

组态软件采用类似资源浏览器的窗口结构，并对工业控制系统中的各种资源（设备、标签量和画面等）进行配置和编辑；处理数据报警和系统报警；提供多种数据驱动程序；各类报表的生成和打印输出；使用脚本语言提供二次开发功能；存储历史数据，并支持历史数据的查询等。

3．组态软件的发展趋势

新技术在组态软件中的应用，使得组态软件呈现如下发展趋势：

1）多数组态软件提供多种数据采集驱动程序（driver），用户可以进行配置。驱动程序通常由组态软件开发商提供，并按照某种规范编写。

2）脚本语言是扩充组态系统功能的重要手段。脚本语言大体有两种形式：一是 C/BASIC 语言；二是微软的 VBA 编程语言。

3）具备二次开发的能力。在不改变原来系统的情况下，向系统增加新功能的能力。增加新功能最常用的就是 ActiveX 组件的应用。

4）组态软件的应用具有高度的开放性。

5）与 MES（Manufacturing Execution System）和 ERP（Enterprise Resource Planning）系统紧密集成。

6）现代企业的生产已经趋向国际化、分布式的生产方式。互联网是实现分布式生产的基础。组态软件将原来的局域网运行方式跨越到支持 Internet。

4．常用的组态软件简介

1）InTouch，它是最早进入我国的组态软件。早期的版本采用 DDE 方式（动态数据交换）与驱动程序通信，性能较差。新的版本采用了 32 位 Windows 平台，并提供 OPC 支持。

2）iFIX，它是 Intellution 公司起家时开发的软件，先后被爱默生公司、GE 公司收购。iFIX 的功能强大，使用比较复杂。iFIX 驱动程序和 OPC 组件需要单独购买。iFIX 的价格也比较贵。iFIX 的用户众多，目前全球的用户已经超过 20 万个。

3）Citech，澳大利亚 CiT 公司的 Citech 是较早进入中国市场的产品。Citech 的优点是：操作方式简洁，但脚本语言比较麻烦，不易掌握。

4）WinCC，SIEMENS 公司的 WinCC 是后起之秀，1996 年才进入市场，当年就被美国的 Control Engineering 杂志评为当年的最佳 HMI 软件。它是一套完备的组态开发环境，内嵌 OPC。WinCC6.0 采用 Microsoft SQLServer 2000 数据库进行生产数据存档，同时它具有 Web 服务器功能。

5）组态王，组态王是北京亚控公司的产品，是国产组态软件的代表，在国内有一定的市场。组态王提供了资源管理器式的操作界面，并且提供以汉字为关键字的脚本语言支持，这点是国外组态软件很难做到的。另外，组态王提供了丰富的国内外硬件设备驱动程序，这点国外知名组态软件也很难做到。

6）三维力控，三维力控是国内较早开发成功的组态软件，其最大的特点就是基于真正意义的分布式实时数据库的三层结构，而且实时数据库是可组态的。三维力控组态软件也提供了丰富的国内外硬件设备驱动程序。

另外，国内外比较有名的组态软件还有 GE 的 Cimplicity、华富计算机公司的开物和北京昆仑通态的 MCGS 等。总之，在国内，一般比较大型的控制系统多用国外的组态软件，而在中低端市场，国产组态软件则有一定的优势。

5．组态王KINGVIEW6.5.1 的系统要求

（1）硬件

奔腾 PIII 800 以上 IBM PC 或兼容机。

（2）内存

最少 128MB，推荐 256MB。

（3）显示器

VGA、SVGA 或支持桌面操作系统的任何图形适配器。要求最少显示 256 色。

（4）鼠标



任何 PC 兼容鼠标。

（5）通信

RS-232C。有的电脑没有 RS-232C 接口也可用 USB- RS232C 转换器转换后，代替 RS-232C 接口使用。

（6）并行口或 USB 口

用于插入组态王加密锁。

（7）操作系统

Win2000/WinNT4.0（补丁 6）/Win XP 简体中文版。

6．安装KINGVIEW6.5.1 组态软件

注意安装完成组态软件后，最好将组态软件解密狗插在并口上（若为 USB 口则插在 USB 接口上），否则组态软件只能进入演示方式，两小时必须重新启动程序，而且在此状态下，用户最多只能使用 64 点，用户能使用的变量就非常有限，很容易不够用。

【关键点】低版本的解密狗对高版本是不起作用的，例如读者购买的 6.51 版的解密狗对 6.53 版的组态王软件是不起作用的。

9.1.2 建立工程

建立一个工程监视搬运站“开始”带灯按钮的明暗状态和“手动/自动”旋钮的位置。计算机为台式计算机，配置满足要求，搬运站的 PLC 为 CPU 314C-2DP。

1．将搬运站的程序下载到PLC中

如图 9-1 所示的梯形图表明“按钮”灯的明暗由 Q0.0 决定。假设 PLC 的地址为 2（一般默认为 2）。

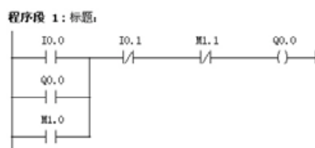


图 9-1　程序

2．连接通信电缆

用 MPI 电缆（若一端为 RS-232C 接口）将计算机的 RS-232C 串口与 PLC（CPU 314C-2DP）的编程口（MPI 口）连接在一起。

3．建立工程的过程

1）双击“组态王”图标，弹出工程管理器，如图 9-2 所示。



图 9-2　工程管理器

2）单击"新建"图标，弹出"新建工程向导之一"对话框，如图 9-3 所示。

3）在"新建工程向导之一"对话框中，单击"下一步"按钮，弹出"新建工程向导之二"对话框，如图 9-4 所示，单击"浏览"按钮，指定新建工程存储的路径。



图 9-3　新建工程向导之一

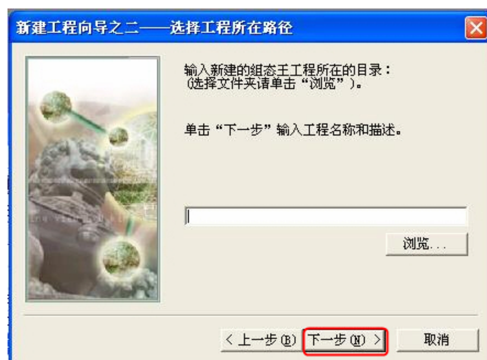


图 9-4　新建工程向导之二

4）单击"下一步"按钮，弹出"新建工程向导之三"对话框，在对话框的输入工程名称中输入"搬运站"，在工程描述中输入"示范工程"，如图 9-5 所示，再单击"完成"按钮。工程名称和工程描述并无特殊规定。

5）接着弹出"新建组态王工程"，单击"是"按钮，将工程设置为当前工程，如图 9-6 所示。



图 9-5　新建工程向导之三

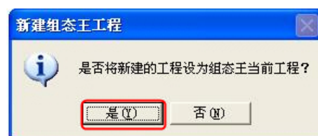


图 9-6　新建组态王工程

6）单击"开发"按钮，系统进入工程浏览器界面，如图 9-7 所示。

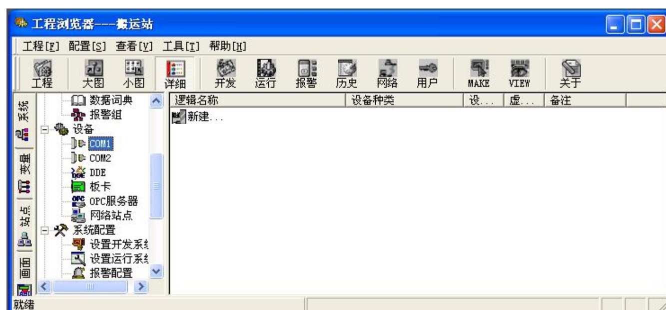


图 9-7　工程浏览器（1）

7）在工程浏览器的左侧选中“COM1”，再双击右侧的“新建”新建...，弹出“设备配置向导——生产厂家、设备名称、通信方式”对话框，如图 9-8 所示。

8）展开“PLC”选定“西门子”下“S7-300 系列”的“MPI”通信方式，如图 9-9 所示，再单击“下一步”按钮，弹出“设备配置向导——逻辑名称”对话框，如图 9-10 所示。



图 9-8 设备配置向导——生产厂家、设备名称、通信方式（1）



图 9-9 设备配置向导——生产厂家、设备名称、通信方式（2）

9）为外部设备取逻辑名称为“S7300”，名称可以是汉字，如图 9-10 所示。单击“下一步”按钮，弹出“设备配置向导——选择串口号”对话框，如图 9-11 所示。



图 9-10 设备配置向导——逻辑名称

10）选择串口为“COM1”，单击“下一步”按钮，弹出“设备配置向导——设备地址指南”对话框，如图 9-12 所示，将地址修改为“2.2”。设备地址格式：A.B。其中，A 指 PLC 在 MPI 总线上的地址，取值范围为 0～126；B 指 PLC 的 CPU 在 PLC 机架上的槽号，取值范围为 0-126。例如：2.2 是指 PLC 在 MPI 总线上的地址为 2，PLC 的 CPU 占用的槽号为 2。

【关键点】注意，这个地址必须与下载 PLC 程序时设置的地址完全一致，否则，通信是

不能建立的。如果下载 PLC 程序时，设置的地址是 6，那么，这个地方地址也应该是 6.2。

11）单击图 9-12 中的“下一步”按钮，弹出“通信参数”对话框，如图 9-13 所示，不改变参数，单击“下一步”按钮，弹出“设备安装向导——信息总结”对话框，如图 9-14 所示，单击“完成”按钮。至此设备的定义完成。



图 9-11　设备配置向导——选择串口号



图 9-12　设备配置向导——设备地址指南



图 9-13　通信参数



图 9-14　设备安装向导——信息总结

12）回到“工程浏览器”界面，选定左侧的“数据词典”，如图 9-15 所示。双击“新建”，弹出“定义变量”对话框，如图 9-16 所示。作如下设置：

变量名：灯。

变量类型：I/O 离散（就是与外部设备有关的离散量）。

连接设备：S7300。

寄存器：A0.0（组态王中的 A0.0 实际上就是 PLC 的 Q0.0）。

数据类型：Bit。

读写属性：读写。

再单击“确定”按钮。





图 9-15　工程浏览器——新建变量

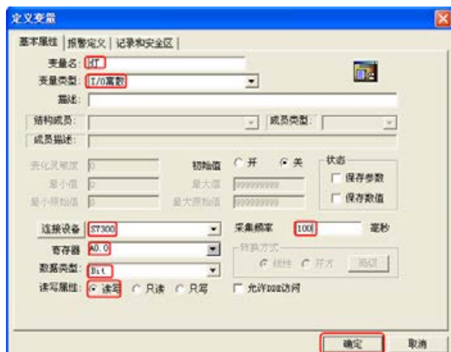
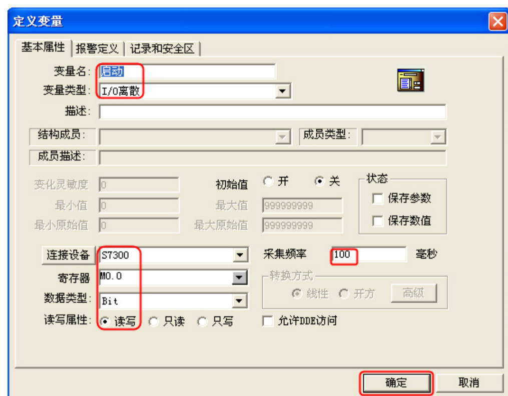


图 9-16　定义变量（1）

13）重复 12）的步骤，作如图 9-17 的设置。至此变量定义完毕。



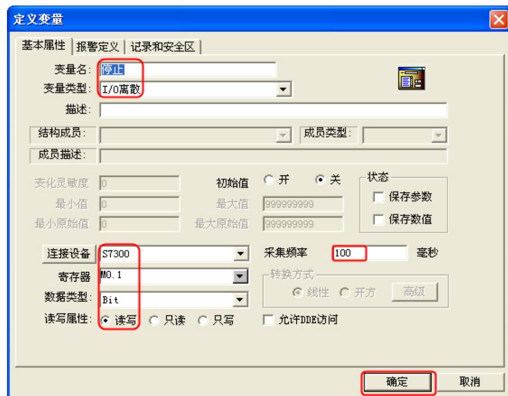


图 9-17　定义变量（2）

14）回到“工程浏览器”，选定左侧的“画面”，如图 9-18 所示。再单击“新建”按钮，弹出“新画面”对话框，在画面名称中输入“xxh”。再单击“确定”按钮，如图 9-19 所示。

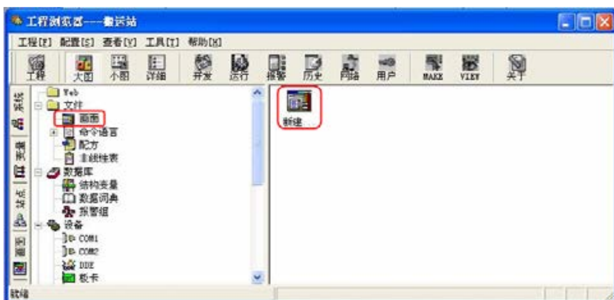


图 9-18 工程浏览器



图 9-19　新画面

15）打开图库，操作方法如图 9-20 所示。

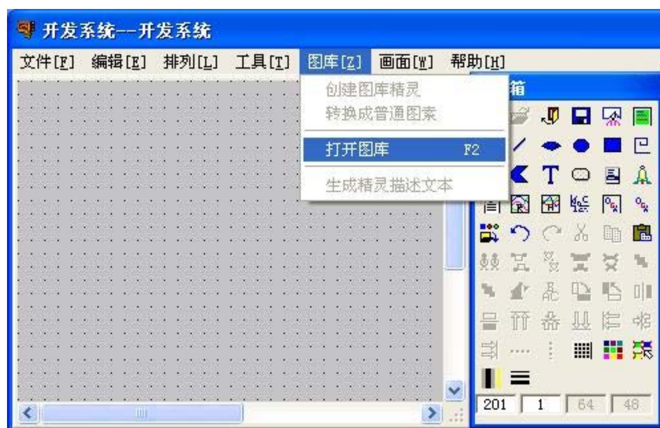


图 9-20　打开图库

16）在图库的按钮中，双击如图 9-21 所示的方形按钮，再在画面中，按住鼠标的左键拖出方形的带灯按钮，用同样的方法在图库的开关中，双击如图 9-21 所示的旋钮式按钮，再在画面中，按住鼠标的左键拖出旋钮式按钮。至此，画面已经制作完成。

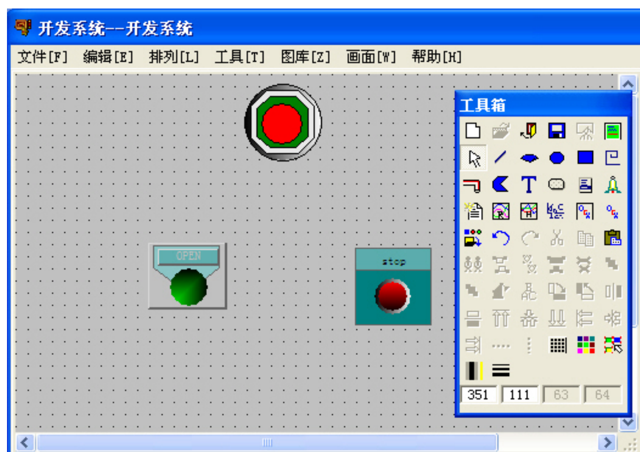


图 9-21　画面

17）以下的操作为动画连接。双击如图 9-21 画面中的按钮（OPEN），弹出“按钮向导”对话框，在此对话框中单击变量名右边的“？”，弹出“选择变量名”对话框，如图 9-22 所示，选定事先定义的变量“启动”，单击“确定”按钮即可。同样设置按钮（STOP）为“停止”。

18）双击如图 9-21 画面中的指示灯，弹出“指示灯向导”对话框，在此对话框中单击变量名右边的“？”，弹出“选择变量名”对话框，选定事先定义的变量“灯”，单击“确定”按钮即可。至此，动画连接已经完成。

19）单击“文件”菜单，再单击“全部保存”子菜单，整个工程被保存，如图 9-23 所示。



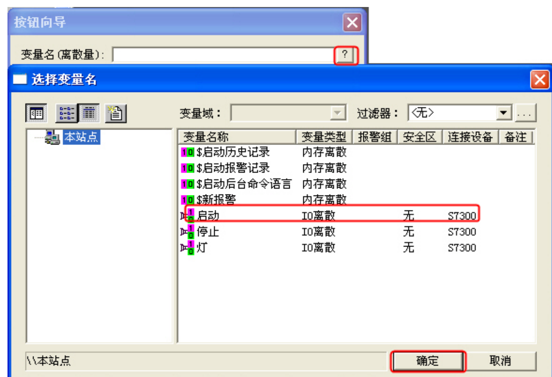


图 9-22　动画连接

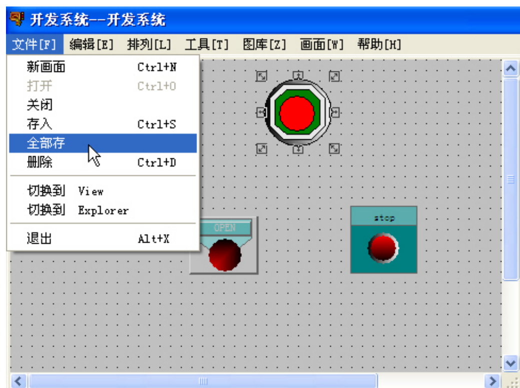


图 9-23　保存工程

20）回到工程浏览器，双击“设置运行系统”图标，弹出“运行系统设置”对话框，选中画面“xxh”作为主画面，再单击“确定”按钮即可，如图 9-24 所示。



图 9-24　设置运行系统

21）将 PLC 的外围电路连接完毕，接通 PLC 的电源。组态软件回到组态王工程管理器，单击“运行”按钮，如图 9-25 所示。



图 9-25　组态王工程管理器

22）运行时的界面如图 9-26 所示。可以看到：当按下左边的按钮灯亮，当按下右边的按钮灯灭。

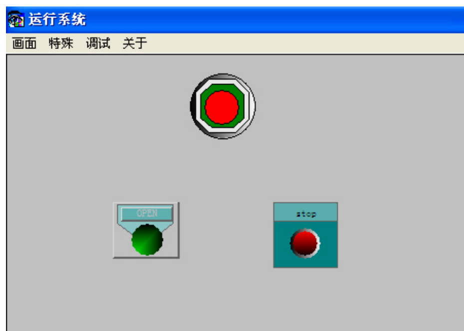


图 9-26　组态王运行界面

9.2　搬运站组态工程的建立

9.2.1　变量

1．基本变量类型

变量的基本类型共有两类：内存变量、I/O 变量。I/O 变量是指可与外部数据采集程序直接进行数据交换的变量，如下位机数据采集设备（如 PLC、仪表等）或其他应用程序（如 DDE、OPC 服务器等）。这种数据交换是双向的、动态的，就是说：在“组态王”系统运行过程中，每当 I/O 变量的值改变时，该值就会自动写入下位机或其他应用程序；每当下位机或应用程序中的值改变时，“组态王”系统中的变量值也会自动更新。所以，那些从下位机采集来的数据、发送给下位机的指令（如“反应罐液位”、“电源开关”等变量），都需要设置成“I/O 变量”。

内存变量是指那些不需要和其他应用程序交换数据，也不需要从下位机得到数据，只在“组态王”内需要的变量（如计算过程的中间变量），就可以设置成“内存变量”。

2．变量的数据类型

组态王中变量的数据类型与一般程序设计语言中的变量比较类似，主要有以下几种：

（1）实型变量

类似一般程序设计语言中的浮点型变量，用于表示浮点（float）型数据，取值范围 10E-38～10E+38，有效值 7 位。

（2）离散变量

类似一般程序设语言中的布尔（BOOL）变量，只有 0、1 两种取值，用于表示一些开关量。

（3）字符串型变量

类似一般程序设计语言中的字符串变量，可用于记录一些有特定含义的字符串，如名称、密码等，该类型变量可以进行比较运算和赋值运算。字符串长度最大值为 128 个字符。

（4）整数变量

类似一般程序设计语言中的有符号长整数型变量，用于表示带符号的整型数据，取值范围-2147483648～2147483647。

（5）结构变量

当组态王工程中定义了结构变量时，在变量类型的下拉列表框中会自动列出已定义的结



构变量，一个结构变量作为一种变量类型，结构变量下可包含多个成员，每一个成员就是一个基本变量，成员类型可以为：内存离散、内存整型、内存实型、内存字符串、I/O 离散、I/O 整型、I/O 实型和 I/O 字符串。

9.2.2 动画相关

1．连接概述

工程人员在组态王开发系统中制作的画面都是静态的，那么它们如何才能反映工业现场的状况呢？这就需要通过实时数据库，因为只有数据库中的变量才是与现场状况同步变化的。数据库变量的变化又如何导致画面的动画效果呢？通过“动画连接”。“动画连接”就是建立画面的图素与数据库变量的对应关系。这样，工业现场的数据，比如温度、液面高度等，当它们发生变化时，通过 I/O 接口，将引起实时数据库中变量的变化，如果设计者曾经定义了一个与这个变量相关画面图素，比如指针，我们将会看到指针也同步偏转。

动画连接的引入是设计人机接口的一次突破，它把工程人员从重复的图形编程中解放出来，为工程人员提供了标准的工业控制图形界面，并且由可编程的命令语言连接来增强图形界面的功能。图形对象与变量之间有丰富的连接类型，给工程人员设计图形界面提供了极大的方便。“组态王”系统还为部分动画连接的图形对象设置了访问权限，这对于保障系统的安全具有重要的意义。

图形对象可以按动画连接的要求改变颜色、尺寸、位置、填充百分数等，一个图形对象又可以同时定义多个连接。把这些动画连接组合起来，应用程序将呈现出令人难以想象的图形动画效果。

2．填充属性连接

填充属性连接使图形对象的填充颜色和填充类型随连接表达式的值而改变，通过定义一些分段点（包括阈值和对应填充属性），使图形对象的填充属性在一段数值内为指定值。

填充属性动画连接的设置方法为：在“动画连接”对话框中选择“填充属性”按钮，弹出的对话框（如图 9-27 所示）各项意义如下：

表达式：用于输入连接表达式，右边的“？”可以查看已定义的变量名和变量域。

本例为封闭圆形对象定义填充属性连接，阈值为 0 时填充属性为红色，阈值为 1 时为蓝色。画面程序运行时，当变量“Q30”的值为 0 时，圆形为红色，表明搬运站没有与下一站通信；变量为 1 时，圆形为蓝色，表明搬运站正在与下一站通信，如图 9-27 所示。

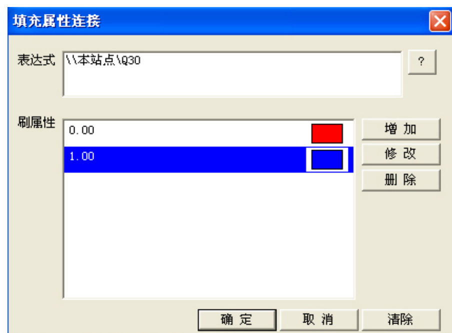


图 9-27 填充属性连接

3．离散值输入连接

离散值输入连接是使被连接对象在运行时为触敏对象，单击此对象后弹出输入值对话框，可在对话框中输入离散值，以改变数据库中某个离散类型变量的值。离散值输入如图 9-28 所示。

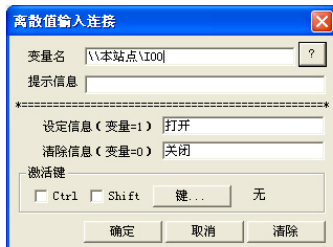


图 9-28　离散值输入连接

4．水平移动连接

水平移动连接是使被连接对象在画面中随连接表达式值的改变而水平移动。移动距离以像素为单位，以被连接对象在画面制作系统中的原始位置为参考基准。水平移动连接常用来表示图形对象实际的水平运动。

水平移动连接的设置方法为：在“动画连接”对话框中单击“水平移动”按钮，弹出“水平移动连接”对话框。水平移动连接如图 9-29。

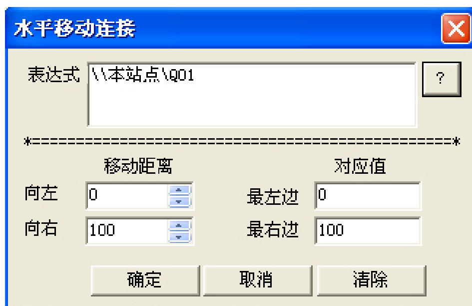


图 9-29　水平移动连接

对话框中各项设置的意义如下：

表达式：在此编辑框内输入合法的连接表达式，单击“？”按钮可查看已定义的变量名和变量域。

向左：输入图素在水平方向向左移动（以被连接对象在画面中的原始位置为参考基准）的距离。

最左边：输入与图素处于最左边时相对应的变量值,当连接表达式的值为对应值时，被连接对象的中心点向左（以原始位置为参考基准）移到最左边规定的位置。

向右：输入图素在水平方向向右移动（以被连接对象在画面中的原始位置为参考基准）的距离。

最右边：输入与图素处于最右边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向右（以原始位置为参考基准）移到最右边规定的位置。

5．隐含连接

隐含连接是使被连接对象根据条件表达式的值而显示或隐含。本例中建立一个表示带灯按钮灯是否“亮”，当变量“Q02”为 1 时，带灯按钮“亮”，绿色圆形不隐含，当变量“Q02”为 0 时，带灯按钮“暗”，绿色圆形隐含。隐含连接如图 9-30。



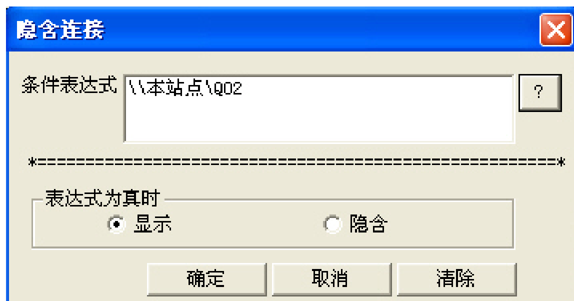


图 9-30　隐含连接

隐含连接的设置方法是：在“动画连接”对话框中单击“隐含”按钮，弹出对话框。

对话框中各项设置的意义是：

条件表达式：输入显示或隐含的条件表达式，单击“？”可以查看已定义的变量名和变量域。

表达式为真时：规定当条件表达式值为 1（TRUE）时，被连接对象是显示还是隐含。如图 9-30 所示，选定“表达式值为真时”显示，所示 Q02=1 时，其连接对象“显示”，否则其连接对象就“隐含”（不显示）。

9.2.3　命令语言程序

1．命令语言类型

组态王中命令语言是一种在语法上类似 C 语言的程序，工程人员可以利用这些程序来增强应用程序的灵活性，处理一些算法和操作等。

命令语言都是靠事件触发执行的，如定时、数据的变化、键盘键的按下、鼠标的点击等。根据事件和功能的不同，包括应用程序命令语言、热键命令语言、事件命令语言、数据改变命令语言、自定义函数命令语言、动画连接命令语言和画面命令语言等。具有完备的词法语法查错功能和丰富的运算符、数学函数、字符串函数、控件函数、SQL 函数和系统函数。各种命令语言通过“命令语言编辑器”编辑输入，在“组态王”运行系统中被编译执行。

其中应用程序命令语言、热键命令语言、事件命令语言、数据改变命令语言可以称为“后台命令语言”，它们的执行不受画面打开与否的限制，只要符合条件就可以执行。另外可以使用运行系统中的菜单“特殊/开始执行后台任务”和“特殊/停止执行后台任务”来控制所有这些命令语言是否执行。而画面和动画连接命令语言的执行不受影响。也可以通过修改系统变量“$启动后台命令语言”的值来实现上述控制，该值置 0 时停止执行，置 1 时开始执行。

2．应用程序命令语言

在工程浏览器的目录显示区，选择“文件”→“命令语言”→“应用程序命令语言”，则在右边的内容显示区出现“请双击这儿进入<应用程序命令语言>对话框…”图标，如图 9-31 所示。

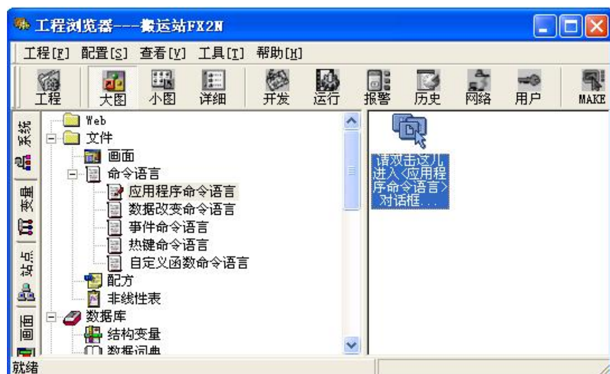


图 9-31　应用程序命令语言

双击“请双击这儿进入<应用程序命令语言>对话框…”图标，则弹出“应用程序命令语言”对话框，如图 9-32 所示。

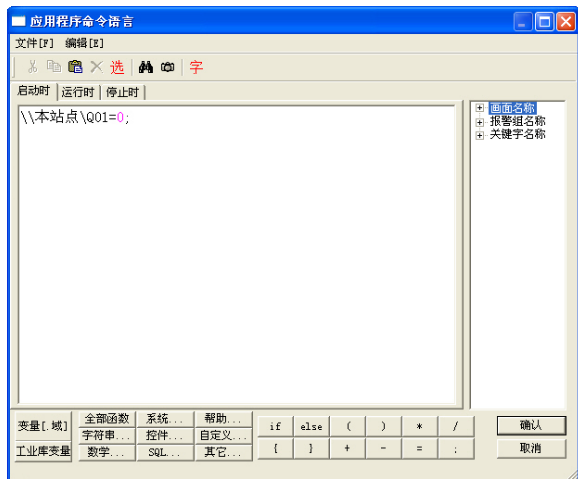


图 9-32　命令语言

3．命令语言语法

命令语言程序的语法与一般 C 程序的语法没有大的区别，每一程序语句的末尾应该用分号“；”结束。在使用 if…else…、while（）等语句时，其程序要用花括号“{ }”括起来。

（1）运算符

用运算符连接变量或常量就可以组成较简单的命令语言语句，如赋值、比较、数学运算等。命令语言中可使用的运算符以及运算符优先级与连接表达式相同。运算符有以下几种：

～　　　取补码，将整型变量变成“2”的补码。

*　　　乘法。

/　　　除法。

％　　　模运算。

＋　　　加法。

－　　　减法（双目）。

&　　　整型量按位与。



| 整型量按位或。
^ 整型量异或。
&& 逻辑与。
|| 逻辑或。
< 小于。
> 大于。
<= 小于或等于。
>= 大于或等于。
= = 等于（判断）。
!= 不等于。
= 等于（赋值）。

（2）赋值语句

赋值语句用得最多，语法如下：

变量（变量的可读写域）＝表达式。

可以给一个变量赋值，也可以给可读写变量的域赋值。

例如：

自动开关=1：表示将自动开关置为开(1 表示开,0 表示关)。

颜色=2：将颜色置为黑色(如果数字 2 代表黑色)。

反应罐温度.priority=3：表示将反应罐温度的报警优先级设为 3。

（3）IF−ELSE 语句

IF-ELSE 语句用于按表达式的状态有条件地执行不同的程序，可以嵌套使用。语法为：

```
IF(表达式)
{
        一条或多条语句;
}
ELSE
{
        一条或多条语句;
}
```

注意：IF-ELSE 语句里如果是单条语句可省略花括弧“{ }”，多条语句必须在一对花括弧“{ }”中，ELSE 分支可以省略。

如：

```
IF (step = = 3)
颜色=“红色”;
```

上述语句表示当变量 step 与数字 3 相等时, 将变量颜色置为“红色”（变量“颜色”为内存字符串变量）

如：

```
IF（出料阀= = 1）
```

```
    出料阀=0; //将离散变量“出料阀”设为 0 状态
ELSE
出料阀=1;
```

上述语句表示将内存离散变量“出料阀”设为相反状态。IF-ELSE 里是单条语句可以省略“{ }”。

如：

```
IF (step= =3)
{
        颜色=“红色”;
        反应罐温度.priority=1;
}
ELSE
{
        颜色=“黑色”;
        反应罐温度.priority=3;
}
```

上述语句表示当变量 step 与数字 3 相等时, 将变量颜色置为“红色”（变量“颜色”为内存字符串变量），反应罐温度的报警优先级设为 1；否则变量颜色置为“黑色”，反应罐温度的报警优先级设为 3。

（4）WHILE（）语句

当 WHILE（）括号中的表达式条件成立时，循环执行后面“{ }”内的程序。语法如下：

```
WHILE(表达式)
{
    一条或多条语句(以；结尾)
}
```

如：

```
WHILE（循环<=10）
{
    ReportSetCellvalue(“实时报表”，循环，1，原料罐液位);
    循环=循环+1;
}
```

当变量“循环”的值小于等于 10 时，向报表第一列的 1～10 行添入变量“原料罐液位”的值。应该注意使 WHILE 表达式条件满足，然后退出循环。

9.2.4　创建搬运站工程

在前面的项目中，读者已经完成了搬运站的安装和调试，因此读者对于搬运站已经非常熟悉，请利用组态王软件建立一个组态工程监控搬运站的工作过程。以下仅仅给出建立这个工程的提示，具体步骤由读者完成。



1．新建一个工程

2．建立组态画面

画面的右上角有计算机上的实时时间；右下角有按钮的开关状态、指示灯的颜色信息和明暗状态显示；右侧的中部则显示主机状态、运行状态和工件信息；左边则是搬运站的外形图，共有 5 张，每一张表示搬运站的一个位置，组态画面如图 9-33 所示；而当组态软件运行时，只显示一张图片，其余四张被隐含，运行时组态画面如图 9-34 所示。当然，组态画面的布置方案可以千差万别，以上仅仅是一种方案，读者不要局限在其中。

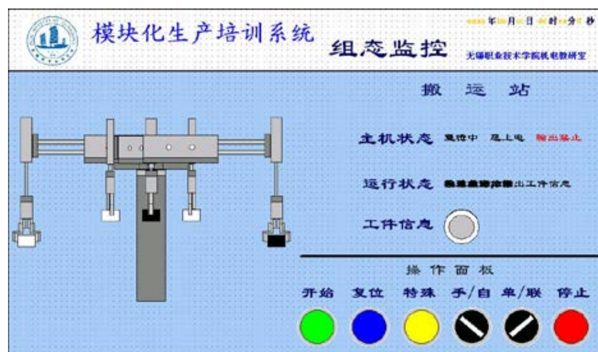


图 9-33　组态画面

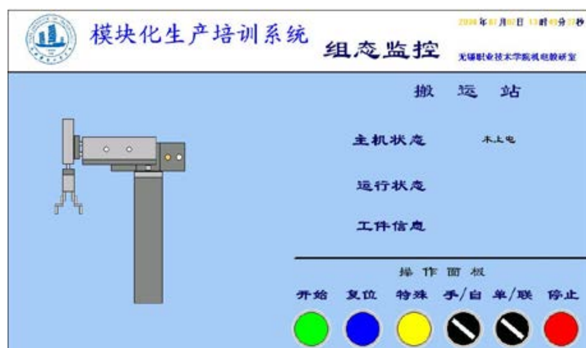


图 9-34　运行时的组态画面

小结

重点难点总结

1．掌握建立一个组态软件工程的一般方法。

2．组态软件中的变量、动画和命令程序。

习题

1．组态软件的功能和发展趋势？

2．国内外还有哪些知名的组态软件？这些知名的组态软件有何特点？

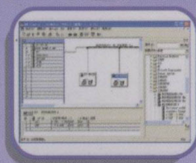
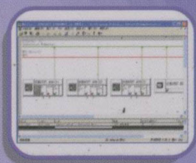
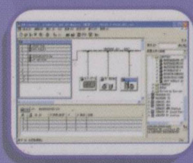
3．用组态王建立一个简单工程大致需要哪些步骤？

参 考 文 献

[1] 向晓汉，陆彬． 西门子 PLC 高级应用实例精解[M]. 北京：机械工业出版社, 2010.
[2] 向晓汉，陆彬. 电气控制与 PLC 技术[M]. 北京：人民邮电出版社，2009.
[3] 刘楷. 深入浅出 西门子 S7-300 PLC [M]. 北京：北京航空航天大学出版社，2004.
[4] 柴瑞娟. 西门子 PLC 编程技术及工程应用[M]. 北京：机械工业出版社，2007.
[5] 张运刚. 从入门到精通－西门子 S7-300/400PLC 技术与应用[M]. 北京：人民邮电出版社，2007.
[6] 秦益霖. 西门子 S7-300PLC 应用技术 [M]. 北京：电子工业出版社，2007.
[7] 廖常初. S7-300/400PLC 应用技术[M]. 北京：机械工业出版社，2005.
[8] 严盈富. 监控组态软件与 PLC 入门[M]. 北京：人民邮电出版社，2006.
[9] 崔坚. 西门子工业网络通信指南[M]. 北京：机械工业出版社，2009.
[10] 崔维群，孙启发. S7-300/400 可编程控制器原理与应用[M]. 北京：北京航空航天大学出版社，2009.
[11] 西门子(中国)有限公司. S7-200 可编程控制器系统手册，2008.
[12] 西门子(中国)有限公司. MICROMASTER 440 标准变频器使用大全，2007.
[13] 周健. 网络技术基础[M]. 北京：人民邮电出版社，1998.
[14] 张春. 深入浅出 西门子 S7-1200 PLC [M]. 北京：北京航空航天大学出版社，2009.
[15] 龚中华. S7-200/300/400PLC 应用技术－提高篇[M]. 北京：人民邮电出版社，2008.

ISBN 978-7-111-32002-9
策　划：时静
封面设计：子时文化 ZiShi Culture

S7-300/400 PLC
基础与案例精选

内容提要

本书从基础和实用出发，涵盖的主要内容包括S7-300/400 PLC、变频器和组态软件。全书分两个部分：第一部分为入门学习，主要介绍S7-300/400 PLC的硬件和接线、STEP-7软件的使用、PLC的编程语言、编程方法与调试；第二部分为实例应用，包括PLC的通信、PLC在过程控制的应用、PLC在变频调速中的应用和上位机对PLC的监控。

本书内容丰富，重点突出，强调知识的实用性，几乎每章中都配有大量实用的例题，便于读者模仿学习，另外每章配有习题供读者练习。大部分实例都有详细的软件、硬件配置清单，并配有接线图和程序。本书的配套资源中有重点内容的程序和操作视频资料。

本书可以作为工程技术人员学习S7-300/400 PLC入门和提高级用书，也可以作为大中专院校机电类、信息类专业的教材。

上架建议　工业技术/电气工程

ISBN 978-7-111-32002-9



地址：北京市百万庄大街22号
电话服务
社服务中心：(010)88361066
销售一部：(010)68326294
销售二部：(010)88379649
读者购书热线：(010)88379203

邮政编码：100037
网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

定价：40.00元